NOV/DEC '87
Vol. 4 No. 1

$3.00
U.S. Funds

Paul M Anderson Jul/88
2321 Caminito Mira
San Diego, CA 92107

BULK RATE
U.S. POSTAGE
PAID
COLTON, OR 97017
PERMIT #51

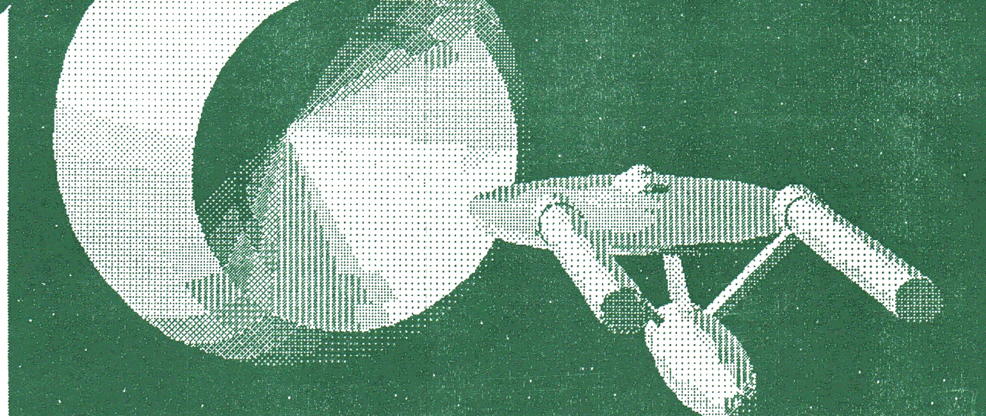
POSTMASTER:
PLEASE EXPEDITE DELIVERY
DATED MATERIAL
ADDRESS CORRECTION REQUESTED

AMERICA'S LARGEST TIMEX SINCLAIR MAGAZINE

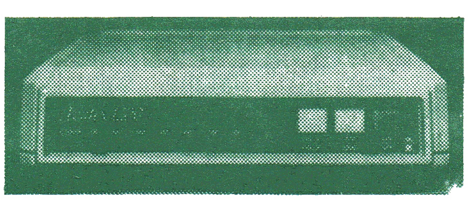
TIME DESIGNS MAGAZINE

3

TDM's Third Anniversary Issue



Focus On TS Telecommunications



CompuServe®
Official Airline Guides
E-Mail

Exclusive First Look At The Z88

SPECTERM-64 (TS-4.1) Terminal Software for T/S2068

You already know that SPECTERM-64 is a terminal software for the T/S2068. You also know that it has a 64 column display (without additional hardware), 1200 baud capability, XMODEM and much, much more. NOW with FULL SIZE Printer Support and EASY INSTALLATION. Works with either the 2050 modem or the Z-SI/O Card (for 1200 baud operation). What you don't know is what your fellow T/S2068 users have to say about it. Here's a few examples:

Dan L., Ohio; "'SUPER' SOFTWARE, GOODBYE MTERM II" * * * * * * * * Norman L., Calif.; ".....Great Product!"

Jim R., Mass.; "Fantastic! 1200 baud, 64 column software! Now all we need is a 1200 baud 2068 BBS program..." (We heard you Jim, look at the TMX-64 software described below.)

Robert S., Calif.; "Great to see all of the display on other BBS's. Like the easy downloading."

SPECTERM-64 (TS-4.1) is available on cassette and also on AERCO FD-68 and JLO SAFE (v2.3 or higher) diskette. Disk versions require NO INSTALLATION. XLATE included FREE of CHARGE. Spectrum version on tape only.

TMX-64 BBS Software for T/S2068 (300/1200 Baud)

This is the "next generation" BBS software for the T/S2068 Personal Color Computer. The TMX-64 BBS software is the first for the T/S2068 that is both 300 and 1200 baud compatible. It is FULL FEATURED, written by Kurt Casby and available only from G & C COMPUTER PRODUCTS.

Z-SI/O RS-232c Serial Interface for T/S2068

Put an RS-232 port on your T/S2068. Use it to drive a printer, plotter or "conventional" RS-232, Hayes compatible modem. SPECTERM-64 and TMX-64 softwares can be used with the Z-SI/O to operate at 1200 baud. JLO, AERCO and STOCK versions available. Fully supported and includes extensive documentation.

ZX-TERM*80 Terminal Software for T/S1000 (ZX81,T/S1500)

Fred Nachbaur wrote this "unbelievable" terminal software for the T/S1000 series computer. Its features are too numerous to list here, but among them are 40, 60 and 80 column display and XMODEM. This is a MUST HAVE for T/S1000 fans. Requires both a NVM (see SCRAM below) and a rampack. HI-RES on the T/S 1000, WOW!

SPECTERM-64 (TS-4.1) on tape.......$30.00 + $2.00 S&H	SPECTERM-64 (TS-4.1) on disk.......$35.00 + $3.00 S&H
TMX-64 BBS on tape................$30.00 + $2.00 S&H	Z-SI/O RS-232c Serial Interface....$79.00 + $3.50 S&H
ZX-TERM*80 (T/S1000 Terminal SW)...$25.00 + $2.00 S&H	SCRAM (Bare Card w/ Docs)..........$12.50 + $2.00 S&H
AVATEX 1200hc (100% Hayes) Modem..$125.00 + $3.50 S&H	Tandon 96TPI Disk Drives (TM101-4).$45.00 + $5.00 S&H
MAGNAVOX (13" RGB Monitor 40).....$255.00 + $13.75 S&H	SEIKOSHA SP-1200 Printer..........$239.00 + $7.50 S&H

*Diskettes * Tapes * Surge Suppressors * Keyboards * Power Supplies * Printer Paper * Mini-Vacs * Disk Cases*

These are but a few of our "stocking stuffers". Send for the complete G & C COMPUTER PRODUCTS Catalog. Call or write for additional info on any product. To order send check or M.O. (COD accepted on most products) to:

Grey & Clifford COMPUTER PRODUCTS
P.O. Box #2186
Inglewood, CA 90305
(213) 759-7406 (orders) or (213) 516-6648 (tech)

Innovative Timex/SINCLAIR Products and QUALITY Computer Supplies

A HAPPY AND SAFE HOLIDAY SEASON
FROM G & C TO YOU AND YOURS

Time Designs Magazine Company
29722 Hult Road
Colton, Oregon 97017
USA
(503) 824-2658
CompuServe ID# 71350,3230

TIME DESIGNS MAGAZINE is published bi-monthly and is Copyright 1987 by the Time Designs Magazine Company, Colton, Oregon 97017. All rights reserved. Reproduction of this magazine in whole or in part by any means without written permission is prohibited by law.

SUBSCRIPTIONS: $16.95 a year for six issues (U.S. funds only), mailed in the U.S. No extra charge for Canada or Mexico (mailed "surface" rate). All other countries please write for information on surface and air mail rates.

CUSTOMER SERVICE: Customer satisfaction is our goal. For subscription service problems, or any questions and comments, please write or call.

CHANGE OF ADDRESS: Please call or write our office if there is any change in your current mailing address to prevent delay or even loss of service. The U.S. Postal Service will not always reliably forward magazines (believe us, not them!). Notify us as soon as you know of any changes.

RENEWAL TIME? To determine your expiration date, simply read the date posted in the upper-right corner of your mailing label (magazine cover). For an example, "Nov/89" would indicate that the November/December 1989 issue would be the last issue received. A form is provided elsewhere to renew your subscription. We also send one notice in case you forget. An early renewal is very much appreciated, and let us know we are doing an adequate job.

DEALERS, HARDWARE/SOFTWARE DEVELOPERS: Write for our display ad rate card, and find out how you can reach the largest number of Timex Sinclair users in North America.

TIME DESIGNS is:

Managing Editor: Tim Woods

Assistant Editor: Stephanie Woods

Production Assistants: D. L. Woods
Don Axmaker
Kim Axmaker

Photography: (unless otherwise noted)
Thomas B. Judd

Printing: Al Underberg and Toad'L Litho Printing & Composition, Oregon City, Oregon 97045.

Frequent Contributors: Joe Williamson, Paul Bingham, Wes Brzozowski, Michael E. Carver, Tim Stoddard, Earl V. Dunnington, Syd Wyncoop, Zack Xavier Haquer, Fred Nachbaur, Mike de Sosa, Joe Newman, Stan Lemke, Duncan Teague, Albert F. Rodriguez, Bill Ferrebee, William C. Andrews, Dick Wagner, Dennis Silvestri, Gale Henslee, M. Vincent Lyon, J. Kevin Paulsen, Warren Fricke, Charles E. Goyette, Kenneth Fracchia, Dennis Jurries, Floyd Chrysler, D. Hutchinson, Herb Bowers, Sr., John McMichael, and others.

International Correspondent: R. Lussier

NOV/DEC '87 Vol. 4 No. 1

TIME DESIGNS MAGAZINE

Information for all models of SINCLAIR, TIMEX, and AMSTRAD personal computers. Serving North America and the International community.

FROM THE EDITOR'S CLUTTERED DESK

Tim Woods

Here We Go Again

With this issue of TIME DESIGNS, we have come to a number of crossroads. Most predominately on my mind, is that we have just wrapped up three years of publishing and about to start our fourth year. Therefore, consider this an "Anniversary Issue".

I want to take a moment to thank everyone who has continued to help out with this magazine. This includes our staff, all of our contributors, our faithful advertisers...and then there is you our readers. If you didn't bother to renew or write notes and letters, we would have certainly "hung up the hat" a long time ago. I may be a bit biased, but I feel this has been a worthwhile project.

Then too, this is the "Holiday Season", so this could also be considered our special Christmas issue. Wherever you celebrate this time of year, may it be a joyous occaison to spend with both family and friends.

This issue is also devoted to TELECOMMUNICATIONS as promised in the previous issue. There are several other items of interest in case this subject isn't your cup of tea. But we will be continuing with our "themed" issues this year, and judging from all of the correspondence I have received, most everyone is in favor of this plan. More on "themed" issues in a moment.

We have also reached another crossroad. Allow me to explain a bit. Back in November of 1984, as we were in the process of putting together the first issue of TIME DESIGNS (a very crude effort), all of us Timex Sinclair users were awaiting the arrival of the Sinclair Quantum Leap computer here in the States. Sinclair Research had already set up their office in Boston, but as you may remember, it was several more months until the computer actually made it here.

But the idea of Sinclair Research actually offering us a brand new Sinclair computer, sort of took away the bad taste that Timex left in our mouth when they decided to quit selling computers.

The rest is history. Sinclair struggled here for a while, suffering from a poor (if not non-existant) marketing strategy, until they packed up and left. Then even Sir Clive sold the major portion of his business to a competitor. That move gave folks even more of an abandoned feeling. Yet because of some enterprising individuals, dealers and user groups, our Timex Sinclair community here in the U.S. has continued on, and still is a fairly strong group compared to other "orphaned" computer lines.

Now the time is November 1987. And just a few weeks ago, the story of stories came accross my desk: SIR CLIVE SINCLAIR IS RETURNING TO THE U.S. to market his new Z88 battery-powered laptop computer. It's true, but one can only speculate how long the "traditional" delay may be before these new machines reach our shores.

The first week of November, Sir Clive himself was here in Las Vegas to attend the FALL COMDEX electronics

QMONIX CODE QL SOFTWARE

Address all correspondence and orders to: OR CALL

QUANTUM COMPUTING, Box 1280, Dover, N.J. 07801 201-328-8846

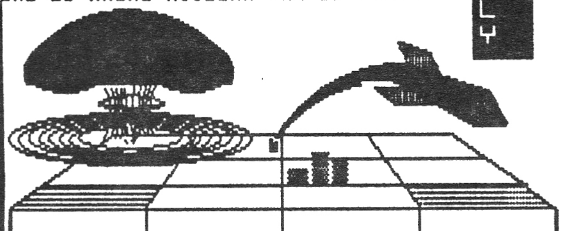
IF FLOPPY THEN SPECIFY DISK SIZE, SECTORS, WHAT ELSE YOUR USING.

GRIDLAND HOLOCOST

SPECIALLY PRICED

HERE IS WHERE NUCLEAR WAR BELONGS:

This is a Sample of the Signon and Winner Graphics Screen as seen with The Game:



INSIDE OF THE COMPUTER!!

GRIDLAND HOLOCOST: Now you can conduct your very own Multiplayer Nuclear War Simulation right within the privacy of your own home! Do it inside of the Computer, right where it belongs. Allows up to 8 player bases. Phenomenal Sounds which can be deselected. Radar graphics window of detonated radiation fields. Uses Ballistic, Probe and Cruse missiles with selectable megaton warheads and sheilding. Excellent for serious interactive hacker group development and general social gatherings. A real nerd buster! Item # G25........$ 9.00

QMONIX COMPUTER CHEF LIBRARY SYSTEM: Using Quill or preferrably The DP EDITOR program, you can now create multiple library volumes of recipes which are sorted and retrieved with keyword lookup system. Also included are custom file configuration utilities and filter programs, which make this a comprehensive recipe file management system. You can taylor your own specific keyword standard list. Nearly two dozen ingredient abbreviations are at your disposal as well. The old recipe card file boxes with all of those cards soiled with ingredients can now be retired - they are history! The QL will print out a disposable copy of the selected recipe as it is needed for use. System requirements: Expanded QL, FLP, Toolkit II, Monitor Mode. Item # S07..........$ 14.00

QMONIX ADDRESS BOOK: An address management application program. Performs address file functions in a simple and easy manner. Instantly searches for any address or phone number in seven different ways. Printouts may be sorted in any of three distinct ways. Full editing and default configuration capabilities. Requires the SuperToolKit II by QJUMP. Item # B10............$ 19.95

Here's an Actual Copy of the QCAL Screen as it appears on The QL:

QCAL

QMONIX CALENDAR

1987			DECEMBER			1987
Sun	Mon	Tue	Wed	Thu	Fri	Sat
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		

Back, Forward or Quit ?

QMONIX QCAL!: Memory Resident Pop-Up Background Calander for the QL. Here's a resident calendar which pops up when you type QCAL. Adding an exclamination mark will maintain it as an independent task which can be switched to at will. QcaL will display the current month automatically or another period can be entered. Hitting the arrow & shift keys will jump up and down by either months or years. Also included is a separate annual calendar generator for printing or displaying a desired 12 month year. A Calendar History file is also provided. Excellent for for continuous time frame reference when composing reports. System Requirements: Ram Disking, memory expansion. Item # U30............ $ 9.00

INTERACTIVE KEYBOARD DRILL SARGENT by QMONIX: Memorize the keyboard and type with greater speed and efficiency. It makes no difference whether you're a novice who never typed before or if you're a professional typist who can do 150 WPM! You will still be equally challenged. WHY? Because the speed of every function of this program will adjust to your ability! There is TOTAL EXTENSIVE USAGE of the QL's Color & Sound capability for an interaction experience that is keyed to your response. With all the bells & whistles you'll forget it's a drill and think you're playing a computer game. Item # S01.......$ 10.00

SPECIAL ONLY UNTIL 12-31-87: ORDER MORE THAN 1 and you can pay just $ 5.00 EACH for ANY of the QMONIX CODE PROGRAMS. Not just those listed here BUT ANY listed in the Catalog & Updates too! ALL QMONIX Programs are supplied in Special Little Color Tinted Plastic Cases Too!! DON'T MISS OUT ON THIS SPECIAL OFFER !!

exposition to show off his new invention to the press and other attendees. "It was the smallest portable at the COMDEX", was the general comment heard, and most of the large U.S. magazines are gearing up to do full write ups in upcoming issues.

CAMBRIDGE COMPUTER LTD is Sir Clive's new company that is responsible for producing the Z88. As you know, about a year ago, after facing tremendous financial problems, Sir Clive sold off the existing Sinclair technology to Amstrad. This included both the Spectrum and the rights to the QL. Currently, Amstrad is marketing a 128K version of the Spectrum, which includes a built-in three inch floppy drive system.

Cambridge Computer has been selling the Z88 for several months now, at first by mail-order, and now in European department stores. But plans for a larger, world-wide market didn't start to materialize until Sir Clive joined forces with the international manufacturer SCI. SCI assembles the Z88 computer in a large factory in Scotland. Since SCI is also based in the U.S., and is a financial backer, the next logical progression was to bring the machine to the U.S. Cambridge Computer has already opened an office in New York.

Many Sinclair fans here in the U.S. all remember the bad business decisions that have followed Sir Clive in recent years. It appeared that the beloved entrepreneur was better in the "brains" department, rather than trying to direct the accounting department. Such ideas as the C5 and the pocket tv have come and gone. Apparently now, the cards are stacked in the right places. Sir Clive has taken a back seat to marketing, letting more aggressive associates handle that end, while he is free to conduct the creation of a variety of new revolutionary products.

Anamartic, another Sinclair Research development company, is at work on a new type of semi-conductor technology called "wafer-scale integration". When fully developed, it will virtually change the electronics field as we know it today. Computers will run faster, more efficient, and will be much more compact in size. The wafer-scale concept has been the dream of the chip industry for years.

Yet another company, Shaye Communications, is at work on another Sinclair concept, a new type of telephone, that will be part light-weight and cordless in the home, and when carried everywhere else, it will utilize cellular phone technology. Current cellular devices are priced at around $1000...this one will reportedly be in the $200 - $300 price range.

In another press interview, Sir Clive announced he is developing a 3D computer display, but wouldn't share any further details on the project.

Hopes are high here for the success of the Z88. It should be well accepted by our Timex Sinclair community, as it retains some of the original Sinclair character and it is a Z80 CPU-based design. Whether it will be a tremendous success, is yet to be seen. The original Sinclair ZX81 sold nearly a million units word-wide, partly due to its low-cost and its surprisingly powerful operating system. If Sir Clive can keep the price down below competition, get the word out, (it already has a powerful operating system...see our first report elsewhere in this issue), then we may see a re-birth of our favorite computer brand.

I would like to thank Bob Howard of W. Covina, CA who using his ham radio talents, was able to procure information about Sir Clive's re-entry into the American computer market, from a friend (a Swedish diplomat) who was recently in London. And also thanks go to Larry Chavarie of Ontario, Canada, who continues to monitor the U.K. press for us.

Seattle Show A Success

On September 26th, well over 100 dedicated Timex Sinclair users got together for the Second Annual North-West TS Mini-Fair, which was held at the Masonic Temple in beautiful Seattle, Washington. The Mini-Fair was co-sponsored by RMG ENTERPRISES and TIME DESIGNS, who were both in attendance as dealers. WEYMIL CORPORATION, another dealer also was in attendance. The members of the Seattle Timex User's Group (SEATUG) were hosts of the one-day event, and did an exceptional job. Three other northwest groups were represented, including VSUG (Vancouver, B.C., Canada), VISTA (Vashon Island, WA), and CCAT/S (Oregon). Some guests attended because of an article in the local paper, "The Seattle Times", and were were genuinely surprised to find out that the Timex Sinclair line was still supported.

The most popular portion of the show were the seven "mini-seminars" given by special guest speakers. Topics and speakers included Vince Lyon (author of "Archive Master") on QL Archive, Syd Wyncoop (TDM columnist) on Z80 Machine Code, Harvey Taylor (program author of Q_LINK) on telecommunications, Michael Carver (a programmer and TDM columnist) on 68000 chip architecture, Wilf Rigter (programmer of ZX81 high-res routines made popular by Fred Nachbaur...also designer of the "Delta Device") on getting maximum use from your ZX81/TS1000, John Scearce (of SEATUG) on TS2068 disk drive systems, and Dick Wagner (CCAT/S N/L editor) on 80 column dot-matrix printers. (Note: RMG Enterprises is offering a video tape of all seven mini-seminars for $15.95. Write to RMG for further info. 1419 1/2 7th Street, Oregon City, OR 97045.)

Next year's Mini-Fair is scheduled to be in the Portland, Oregon area some time near the end of August. Plans include to expand the show to the entire west-coast and surrounding states. Watch TDM for further details.

TS Winterfest

The San Francisco Sinclair show has been cancelled, but the SUNSTATE TIMEX SINCLAIR WINTERFEST is a go! It will be held on March 7-9, in Orlando, Florida at the Orlando Marriott (8001 International Drive, Orlando, FL 32819). The three day show is being sponsored by four user groups from Florida, and is open to all interested Timex Sinclair users. The following TS dealers and companies will be in attendance: Variety Sales, Curry Computer, Quantum Computing, AFR Software, TS2068 Update Newsletter, Aerco, Markel Enterprises, Grey and Clifford Computer Products, Foote Software, Zebra Systems, and Time Designs Magazine. Some international companies have been invited (including Sir Clive!), and two surplus electronic stores from the local area will be there.

Accomodations can be obtained at the convention site hotel, or from one of the surrounding area hotels. The surrounding attractions in the area will be great to take in after the show, including Walt Disney World (just five minutes away). Plan your family vacation now!

Advance tickets can be obtained for $5 single, $9 family (make checks payable to: Northeast Florida T/S Users Group). For complete information, you can write the director: Mary-Lynn Johnson, 249 N. Harden Ave., Orange City, FL 32763; or give the "official" BBS a call at (904) 775-0093. Settings are 8/1/N.

Business

The next issue of TIME DESIGNS will be devoted to the theme of "business". That is, the use of Sinclair's in conducting both personal and professional business. This theme comes as a response to a letter we published in the September/October '87 issue of TDM, in which a reader suggested hearing from others who use their computers for this particular application. This past month, our mailbox has been flooded with letters and cards from those who addressed this issue.

You will want to make sure that your subscription is still current, as you just can't miss this next one! We will have six to eight interviews with individuals, ranging from shop owners, executives, to investment brokers who have passed up the PC "temptation", and instead have utilized their Sinclair hardware and software to conduct their affairs.

In the next issue, Herb Bowers, a retired federal auditor, will return with the all new 1987 FEDERAL INCOME TAX CALCULATOR program and article. He has just received the most updated information, and has completely debugged and extensively checked his program out against the official IRS tables and instructions. Amaze your friends when you actually appear to understand the new tax code!

Bill Ferrebee will show us a simple modification to the old Timex program QUADRA CHART, to make it truly useful. And several other programs/articles will be featured.

The March/April '88 issue is slated for the theme of COMPUTER GRAPHICS. After that, we are wide-open for any suggestions for a theme you would like to see.

Dealers

Quite a number of advertisers in this "pre-Xmas" issue. We hope that you would take the time to read them and even write to these companies (or better yet, order their products!) for catalogs and brochures. They are part of the reason why TIME DESIGNS is still around, and why there is still stuff for your computer.

Great TS User Groups

--check them out!

Have we listed your group's information lately?
Send us the club's name/address for publication.
Get noticed. Bring in new members!

North East Florida T/S Users Group
c/o John Kuhn
1707 King St.
Jacksonville, FL 32204

Greater New Haven Timex Sinclair User Group
c/o Dennis N. Silvestri
256 Lloyd St.
New Haven, CT 06513

New York Timex/Sinclair Enthusiasts
c/o Joe Newman
325 West Jersey St., #2D
Elizabeth, NJ 07202

The old T.O.P.S newsletter for users of the Zebra FDD disk drive system (Timex Portugal), which was edited and produced by Dave Franson, has been resurrected by Ronald Havlen (4307 Chambers Rd., Horseheads, NY 14845).

If anyone has purchased this system and would like to receive further information, we urge you to contact Mr. Havlen. Or, if you know of someone who possesses such a system for their TS2068, please pass this info on to them.

We are quite impressed with the new publication and especially with the regularity with which it comes out. It should greatly enhance communication between Zebra users. Note: the newsletter is for users of both the older (3-piece) system and the newer FDD-3000, and even CP/M users.

HO..HO..HO..HAPPY HOLIDAYS TO ALL FROM CURRY COMPUTER

WE ARE STILL YOUR ONLY "TOTAL" SINCLAIR TIMEX DEALER

We accept Visa and Mastercard. We ship C.O.D.for $2 extra. Pers.checks held 2 weeks.

FOR ZX81 AND TIMEX SINCLAIR 1000

BRAND NEW!!!

TS1000 COMPUTER WITH 16K RAM

PLUS....

CHOOSE 3 PIECES OF SW FROM THE LIST BELOW!!

Croaka-Crawla
Pioneer Trail
Home Asset Mgr.
Keyboard Calcul.
The Coupon Mgr.

$34.95 plus $ 4.00 S+H.

HARDWARE AND BOOKS

Minduare Printer*	$24.95	Ins & Outs of 1000	$ 5.95
16k Ram Pak	19.95	Purful Proj. for T/S	6.95
32K Ram Pak	34.95	Compt I/F Techs	6.95
64K Ram Pak	49.95	VuCalc/VuFile Book	6.95
MemoCalc	24.95	Under. Yr ZX81 ROM	7.95
MemoText	24.95	2068 Basics & Beyond	8.95
HRG I/F	34.95	2068 Beg/Int Guide	8.95
T/S 1000 Compt Pad*.	3.95	2068 Int/Adv Guide	9.95
2040 Paper-3 for	6.95	101 Things u Dead Compt	5.95
Monitor Adapt.	16.95	1000 BASIC Book	3.95

Add .75 s/h per Book unless ordering with s/w.

Take 10% off H/W if ordering w s/w, bks. * supplies limited.

SOFTWARE!!!!!!!!!!!!

ALL 1000 SOFTWARE SOLD 2 FOR $6, 3 FOR $8.50, 4 FOR $10!

Munchees	CroakaCrawler	Forecast/Graph
Pilot	Fin. Rec. Keep.	Invent. Control
Gulp	Geometry I	Space Commando
Red Alert	Alien Invasion	Mix Gm Bag 1
Vu Calc	Organizer	Checkbk Mgr.
Chess	Strat. Football	Hm Improv Plan
Hangman	Nite Gunner	Meteorites
Budgeter	Kybrd Calc	Home Asset Mgr.
Gambler	Flite Simulator	Backgammon
Statistics	Word Test	Labyrinth
2k Games	Supermath	Coupon Manager

AND MUCH, MUCH MORE.....

FOR ORDERS ONLY 24HRS A DAY CALL 1-800-628-2828 EXTENSION 950.

FOR TIMEX SINCLAIR 2068 AND SPECTRUM

ADD A ROMSWITCH TO YOUR 2068 AND YOU CAN RUN SPECTRUM PRGS.

$ 34.95

Gamesmate joystik adpt plugs into cartridge port-allow use of j/s when in Spectrum mode $16.95 .

AND WITH A ROMSWITCH, HERE'S WHAT YOU CAN RUN:

Coin-Op Hits...$13.95 (5 Hits for 1 price)	Paper Boy...$16.95 (Just like Arcade)
Sherlock Holmes......19.95	The Hobbit........19.95
Silent Service.......16.95	Zub...............9.95
Elite Hitpak 6.......17.95	Rebel Star Raider..5.00
Make a Chip..........5.00	Cauldron..........16.95

LOTS OF NEW TITLES COMING!!! CALL.......

★★ Misc. Accessories.. FOR ALL!!!!!

Top Deck Printer Paper Catcher	$38.95
Mini Vac	19.95
Documate (holds paper while typing)	14.95
Datalight	27.95
Mouse Mat	6.95
Monitor Station(allows mon. to suivel)	17.95
Printer stands	8.95up
Top Deck with printer stand	58.95

We sell Printers!!CALL!!

FOR SINCLAIR QL

HARDWARE....

Trump Card - 768k, Disk I/F, TK2, and more	$299.95
Centronics Interface	34.95
RS232 Cable	16.95
RGB Cable (QL end on, Monitor end bare wires)	16.95
Monochrome Monitor Cable	14.95
Modapter (Cable and Terminal Software)	44.95
Modem Cable for 1200 Baud only	16.95
Mitsubishi DSQD 5.25" Single Disk Drive With Case Power and Cable	169.95
Mitsubishi DSQD 5.25" Dual Disk Drive With Case, Power and Cables	289.95
Used 512k RAM (when available) 30 day warranty Depending on conditon...	$89 - $99
Single Disk Drive Cable	19.95
Dual Disk Drive Cable	24.95
Battery Back-up Clock Board	24.95
Avatex 300/1200 Baud Modem	119.95
QL Dust Cover	11.95
Magnavox RGB cable	17.95

Write for QL Parts List..

SOFTWARE

Concept 3D	$29.95	Eye Q	$49.95
Locksmithe	19.95	Desk Top Publisher	119.95
Locksmithe+4Matter.	39.95	Super Media Manager	62.95
Sidewinder - screen dump	19.95	Super Forth	59.95
Toolchest	19.95	The Editor	34.95
Shrivel (128K reset)	12.95	Better Basic	34.95
Metacomco Assember.	44.95	Nucleon	17.95
Presidents	14.95	PC-Import	34.95
Print Driver	9.95	Strip Poker	17.95
QL Mail Merge	24.95	I.C.E.	29.95
Super Checking	14.95	Test Cartr. and loop back cable(diagnostics)	14.95
Wanderer(needs RGB)	24.95	Squadrons	19.95
RamDisk/Print Spool	12.95	QCode terminal s/w	19.95
QDump	9.95	Thomson case (holds 12 cartridges)	2.50
Metacomco Pascal	89.95	OR 5 for	$10.00
Graphic Toolkit	19.95		
Metacomco 'C'	116.95		
Grab Bag	14.95		
Metacomco LISP	79.95		

SHIPPING CHARGES

FREE ON ALL S/W

BOOKS ADD .75 for ea.

HARDWARE CABLES, AND ACCESS. AT UPS GROUND CHARGES TO YOUR ZONE

ROM-Cube $79.95

Allows you to have 4 Rom cartridges ready to use on your QL. Saves wear and tear on port. 5 position switch. 2' ribbon cable.

BOOKS

Inside QL	12.95	QDOS Compan.	12.95
Adv.Progrm.	7.00	WP on QL	7.00
Intr S.B.	7.00	Desktop QL	7.00
Quantum Theory	6.95	Gamesmaster	9.95
Introd.QL	7.95	Adv.Hndbook	6.00
QL SuperBasic	11.95	QL Hndbook	12.95
AI on QL	12.95	DB Mgmt QL	12.95

CALL OR WRITE FOR COMPLETE LIST...

MISC. ITEMS

Psion Organiser II	$179.95 up
Datapaks for Organiser	CALL!!!
Z88 INFO (send $1 for specs and description of machine)	
Flat TV batteries(P500)	6.00 ea.
Magnavox 80 RGB Monitor	299.95
Avatex 300/1200 baud	119.95
Maxell 5.25" DSDD(bx 10) Disks	16.95
Sony 3.5" DSDD(bx 10)	16.95

magazines

Sinclair User, QL World, Computer & Video Games, Atari User, Atari ST User,C64 User....$4 ea or 3 for10.50

WRITE FOR FREE CATALOG

Please specify type of computer you have.

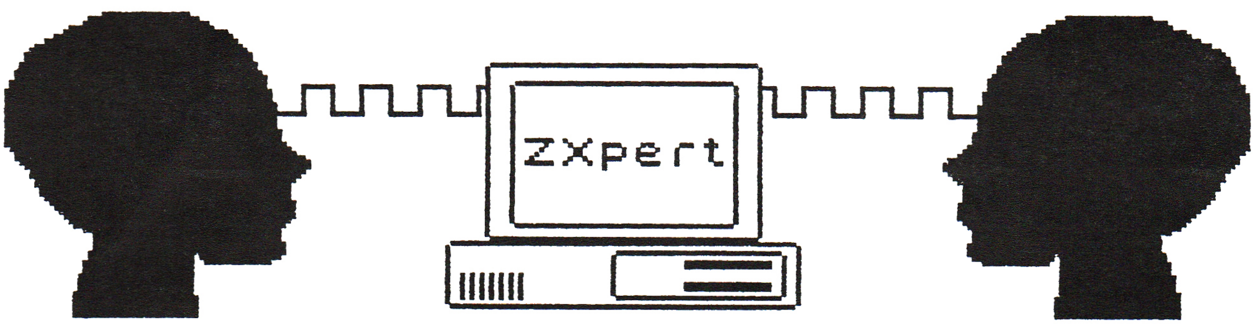
NEW PRODUCTS ANNOUNCEMENT
For TS2068 and SPECTRUM

NovelSoft

brings Artificial Intelligence to the Timex/Sinclair

Introducing...

ZXpert



ZXpert lets you explore the reasoning process of human experts.
With ZXpert you can create an Expert System on any topic you like!

* A useful and educational program
* Easy and fun to use
* No programming language to learn
* On-line help
* Includes 2068 & Spectrum versions
* Supports Microdrives
* Two example knowledge bases
* Superb Manual with Tutorial

plus...

Once you have developed a useful knowledge base, with your permission NovelSoft will sell it and pay you a royalty!!!

Join the revolution with...ZXpert

$19.95 U.S. Plus $3.00 S & H

NovelSoft
A FORMAT FOR THE FUTURE

NEW ADDRESS:
35 Candle Liteway, Willowdale, Ontario, Canada M2R 3J5
phone (416) 665-0290, Compuserve ID# 72477,326

PRODUCTS

For TS2068 and SPECTRUM

Artworx

VERSION 1.1

- Pull-Down Menus
- Several Brushes
- Spray
- Auto-Fill
- Zoom
- Undo
- Several Text Fonts
- Cut & Paste Windows
- Auto-Speed Control
- Magnify & Reduce
- Rotate & Mirror
- Full Attribute Control
- Fully Elastic Shapes including Circle, Box, Triangle, Ray and Line
- Fast Ellipse and Arc
- Includes Spectrum & TS2068 Versions
- Supports Microdrives and Kempston Joystick
- Includes GALLERY, the slide show/animator
- 5 Samples of Artwork
- Excellent Manual

Unshackle your creativity with ... ARTWORX!!

ARTWORX V1.1 - $19.95 U.S. plus $3.00 S & H

REVOLUTIONARY NEW BASIC COMPILER...

TIMACHINE

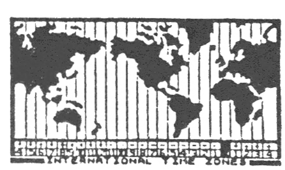
The dream of every BASIC programmer has now been realized!

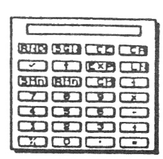
* TIMACHINE will turn your BASIC into super-fast machine code, running up to 200 times faster!
* Handles floating point operations like SIN, COS, TAN
* Handles all BASIC except I/O
* Includes an excellent manual and 4 demonstration programs
* Compiles up to 27K in seconds
* Includes Spectrum & 2068 versions

Super-Charge your BASIC programs with...*TIMACHINE!*

TIMACHINE — $19.95 U.S. Plus $3.00 S & H

THE WORX!





A unique combination of planning aids, decision tools, and utilities.

* Scheduler/Planner
* Notepad
* Telephone Book
* Programmable Calculator
* Decision Factoring
* Real Time Clock
* Biorhythms
* Perpetual Calendar
* International Time Zones
* Superb Manual
* Includes Spectrum & TS2068 Versions
* Supports Microdrives

A TIMELY ADDITION TO YOUR SOFTWARE LIBRARY...THE WORX!

— $19.95 U.S. Plus $3.00 S & H

NovelSoft
A FORMAT FOR THE FUTURE

NEW ADDRESS:
35 Candle Liteway, Willowdale, Ontario, Canada M2R 3J5
phone (416) 665-0290, Compuserve ID# 72477,326

IN THE MAILBAG

Need For Diagnostic Software

[In the July/August issue] you solicited replies from your readers who could repair our computers, "even as a hobby" you said. It is my opinion that we have within our "community" the talent to write diagnostic software to aid in that repair. Hewlett Packard move over. Repair with such an aid is almost a sure thing. I bet one or more writers in collaboration could write a program using graphics that would have details including component placement that would require the "repairman" have only good soldering skills. I bet this program could even run on a 1000 that a 2068 owner had all but disgarded. Because, if we can't keep 'em running, they will just die on our desks. But, being able to repair them will keep them running virtually forever! And the longer they live, the longer their support lives.

John J. Shepard III
Coldwater, Mississippi

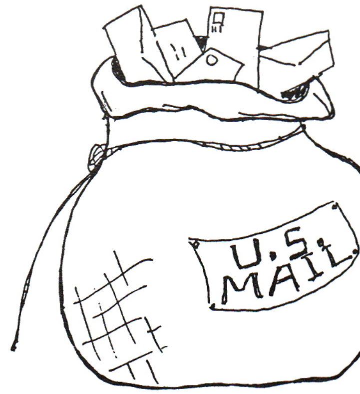
Editor: Bravo, John! A very good idea. And even if there are those that don't prefer to solder, they could at least be more knowledgeable when discussing the problem with the repairman. A sophisticated diagnostic program could even "self-test" areas like the ROM and the RAM to pinpoint the trouble. Color patterns generated on the TS2068 could even help fine tune the video circuitry. And of course if anyone does come up with a program such as this, they are welcome to submit it to TIME DESIGNS for possible publication.

Spectrum User Shares Programs

I was obliged to return to the U.S. after spending several years in Barcelona, where at a language school I worked for, we used Spectrums extensively. They lend themselves wonderfully and inexpensively to programs I was writing, dealing with all phases of the English language. I brought a Spectrum Plus back home with me, unaware that there were enough of them in this country to justify a magazine and software companies, so I've been happily surprised.

I'd like to offer a couple of little utility programs I was able to acquire, ones which I used frequently. The first one is used to put in the unerasable Line 0, useful for adding a copyright notice or anything else, and once in the program, it's there for good, unless an advanced hacker comes along and figures out how to erase it. When you have entered the desired legend in Line 2, RUN it. You will then have Line 0 with this same info. Now erase Line 1 and you have your permanent legend.

The second listing is a program that will erase large blocks of listings, even entire programs, in a split second, almost before you get your finger off the ENTER key. I once bought one of those commercial "tool kit" programs with a DELETE feature, but if you had to erase 100 lines of listing, you could go out for a leisurely cup of coffee while waiting. This little program does it instantaneously. I chose to use Lines 9960 to 9990 only because I never used them for anything



else. You can use any line numbers you wish, making the necessary internal changes in GOTO's. Now, of course, simply MERGE this into the program you plan to modify. The program erases any portion of a listing you want, even itself, but one caution: if there are any UDG's in the program, they will become distorted with each use of them. This is no cause for concern. Once the revised listing has been SAVEd and the Spectrum is unplugged briefly, no problem. It's just that this delete program leaves a residue of oddball things in RAM that have to be cleared out completely, not just NEW'ed. If another program is loaded without completely emptying the memory this residue will play undesirable tricks with the UDG's if any are present.

I hope a few people might find these items useful, and it's nice to know that the Spectrum lives on here in the U.S.

Randall E. Larson
Tucson, Arizona

"NO-DELETE" PROGRAM:

```
1 LET a=PEEK 23637+256*PEEK 23638: POKE a,0: POKE a+1,0: STOP
2 REM Any message you want here after the REM
```

BLOCK DELETION PROGRAM

```
9960 REM DELETE; after using, clear memory completely
9965 CLEAR 65499: RESTORE 9975: LET a=65500
9968 READ n: IF n=257 THEN  GO TO 9980
9970 POKE a,n
9973 LET a=a+1: GO TO 9968
9975 DATA 33,0,0,229,33,0,0,35,205,110,25,227,205,110,25
9978 DATA 32,1,235,225,167,237,82,216,25,205,229,25,201,257
9980 CLS : PRINT "Start :";: INPUT a: PRINT a
9983 PRINT "End   :";: INPUT b: PRINT b
9985 POKE 65501,a-256*INT (a/256): POKE 65502,INT (a/256)
9988 POKE 65505,b-256*INT (b/256): POKE 65506,INT (b/256)
9990 RANDOMIZE USR 65500
```

QL Bar Code Reader anyone?

Does anyone know where one could obtain a bar code reader program (and interface) for the Sinclair QL computer?

Joe E. Jenkins
3100 Mockingbird
Amarillo, TX 79109

Public Domain Program

I am writing to announce that I am putting one of my programs, T/S GRADER for the 2068 in the public domain. I am busy with other projects and see little purpose in extensive advertising for such a limited market. On the other hand, it is the best program of its type that I have seen (I wrote it because I was dissatisfied with other similar programs) and I want any interested persons to be able to obtain it. Therefore, any owners are free to give copies of the program and documentation to friends, user groups, etc. with no obligation to me.

It features machine code speed so you can see any student record or class record in about a second. It handles up to 255 classes simultaneously, each class can have up to 46 students, and each student can have up to 66 grades. For example, you could have 303 students, each with 66 grades, and divided into 7 classes. Or you may prefer something like 1275 students, each with 5 grades, divided into 28 classes. Students can be added to any class, deleted, or transferred between classes. Each class can be alphabetized. Grades are numeric between 0 and 254, each class can have grades weighted separately, bonus and demerit points can be given, and you can do statistical analysis of the grades. You can also calculate semester or yearly averages. Missed tests need not be averaged in.

There are many more features in this $20 program. If someone can not find a friend to make a free copy, I will still sell it for $15 since it is a lot of trouble for me to make individual copies of the documentation. I really prefer that they get it elsewhere if possible.

Robert C. Fischer
804 Old York Hwy, North
Apt 3-B
Dunlap, TN 37327

Christmas Greeting

A friend of mine has sent an interesting computer program that was used on some Christmas cards that he sent out. The mathematics involved shows a great deal of thought and experimentation to achieve the desired result. Listing #1 will work only on a computer that uses the standard ASCII character set, such as the TS2068. The program was written by Bob Lodge of Seattle, WA. Listing #2 is a version that has been written for the TS1000 which does not use the standard ASCII character set.

Thank you for publishing such a fine and unique magazine. I hope to see more "quick and easy" programs like the "Just For Fun" department in the defunct SYNC magazine.

Eric J. Kongs
Wilmington, California

Listing #1

```
10 REM GREETINGS
20 REM BOB LODGE, 1986
30 FOR A=1 TO 2
40 FOR B=1 TO 4
50 LET X=2-ABS(SGN(B-3))
60 FOR C=1 TO X
70 PRINT CHR$(84-7*A+5*B-8*X);
80 NEXT C:NEXT B
90 PRINT CHR$(A+31);
100 NEXT A
```

Listing #2

```
10 FOR A=1 TO 2
20 FOR B=1 TO 4
30 LET X=2-ABS(SGN(B-3))
40 FOR C=1 TO X
60 PRINT CHR$(57-7*A+5*B-8*X);
70 NEXT C
80 NEXT B
90 PRINT CHR$ (27 AND A=2);
100 NEXT A
```

"Get Lucky" Revisited

In reference to the article by Bill Ward on page 7 [of the September/October issue], I have enclosed a program listing created on the TS1000 for producing numbers for the California lottery. This lottery requires six numbers from 1 to 49, with no repeats. I have not yet had enough nerve to try any of my numbers!

Love the mag, except there is too much non-T/S 1000 stuff!

Chapman Burk
El Cerrito, California

Editor: This lottery thing is really catching on. Next issue we've got one for the Ohio lottery and a revised listing for the one in Florida. Good to see the TS1000 in on the action. I won't even attempt to touch that last comment. We have equal number of TS2068 people who say there is too much non-2068 stuff, the QL people feel there is too much non-QL stuff, and perhaps the Z88 owners will feel the same. Seriously though, we are trying to offer a "balanced" publication.

```
   1 REM * DRAWS WINNING NUMBERS
 FOR LOTTERY *
   3 REM * 10 SETS ARE DRAWN *
  10 RAND 0
  40 PRINT "10 SETS OF LOTTERY W
INNERS ARE:"
  70 PRINT
  80 LET E=49
  90 FOR F=1 TO 10
 100 PRINT F;".";
 110 DIM A(6)
 120 FOR N=1 TO 6
 130 LET T=INT (RND*E)
 135 REM * DELETE ZEROES *
 140 IF T=0 THEN GOTO 260
 145 REM * DELETE DUPLICATE NOS.
*
 150 LET A(N)=T
 160 FOR M=1 TO N-1
 170 IF T=A(M) THEN GOTO 260
 180 NEXT M
 190 PRINT TAB (N*4+2);T;",";
 200 NEXT N
 210 PRINT
 220 PRINT
 230 NEXT F
 235 PAUSE 200
 240 COPY
 250 STOP
 260 LET N=N-1
 270 GOTO 200
```

WHICH SYSTEM DO YOU USE?

Reader Tips and Hints for Mass-Storage

Send us your tips, hints, short program listings or whatever you might have that pertains to disk drives, microdrives, wafadrives, even cassettes, and we will print as many as possible. This is NOT a TS2068 column only. So TS1000 mass storage system owners get your printers warmed up to, and send in your tips. Spectrum and QL owners too!

RAMEX MILLENIA K DISK DRIVE SYSTEM (SPDOS)

AUTO FILES
An Under Utilized Facility

Munson H. Cockayne Jr.

Recently, I wrote a letter to TDM about Millenia-K users, and received many replies from RAMEX and SPDOS users alike. Many replies and many questions. I tried to answer all the questions on an individual basis, but experience tells me that the replies were just the "tip of the iceburg".

Some of the common problems were:

1. Converting files to run from disk.
2. Not knowing what "Sequential File" means.
3. Inability to find out more about the system.

Along with these direct questions, I detected a distinct under-utilization of the AUTO facility that is available with SPDOS and other disk systems for the TS2068. I hope to have the opportunity to present some of the other subjects at a later date. This time I want to cover some of the ways AUTO can be used to your advantage.

The manual states that a BASIC program saved with the name "AUTO", will be automatically loaded upon initialization of the system. If this BASIC program is saved with a LINE on which to start, you will have a disk which automatically loads your chosen program. This is a great way to create dedicated disks that can enable others in your household, less computer literate than you, to make use of the computer. I have a disk set up with OMNICALC II as an AUTO file that my wife uses to keep the household budget straight.

An even better use might be to make an AUTO file from a MC program. The manual states that a MC file will be automatically loaded and execution will proceed from the first byte of that file. Now you can make your printer interface software automatically load! "My interface MC doesn't start execution at the first byte of the code." Neither did mine!

I use a Tasman Parallel Interface (modified to a "C" revision as detailed in the user manual). From a look in the Tasman information sheet, "tasintcode" runs from 64716 for a total of 652 bytes. To initialize I call 64719. To make it work right with the disk AUTO function, I counted back three bytes from the start. This gave me 64713. Next I looked up the code for an absolute jump (JP) in MC which is 195 decimal (page 244 of the TS2068 users manual), I calculated the high byte of the address 64719 as 252 (INT (64719/256)), and I calculated the low byte as 207 (64719-256*252). With this information and the code loaded into protected memory, I POKEd 64713,195; 64714,207; 64715,252 (instruction; low byte; high byte). Now all I had to do was PRINT #4; SAVE "AUTO" CODE 64713,655 (652 plus the three bytes I added). Executing a NEW resulted in an "ok" report. I use this AUTO printer routine on my development disks. That way the printer is automatically ready for LISTing and PRINTing anything I want.

In addition to using this file as an AUTO file, it can be renamed with a single character name (maybe "p" for printer). When this file is pulled up with the optional syntax, PRINT #4'p, the MC will AUTO execute. The short program in this article demonstrates this, but more on that in another article.

One other way to make the system more automatic is to use a program to drive the system. This concept is called making a shell for the DOS. A basic program, like the one that follows, is a simple shell if the programs that are loaded by it either reload the shell program or execute a NEW on termination (PRINT #4: LOAD "AUTO" will reload this shell). This program is simple and much more elaborate schemes could be made. Leave the ON ERR statements out if you prefer, but they make the program unstoppable by normal means. (You have to use a symbol shifted "B".)

```
   1 ON ERR RESET: ON ERR GO TO 2000
   2 PRINT #4'p
   3 GO SUB 9000
  10 PRINT "CATALOGUE:"''
  20 FOR I=1 TO TITLES
  30 PRINT I;") ";T$(I)
  40 NEXT I
  50 PRINT #1;"PRESS # OF CHOICE"
  60 PAUSE 0
  70 LET R$=INKEY$
  80 CLS
  90 IF R$="*" THEN ON ERR RESET: STOP
 100 IF R$>"0" AND R$<= STR$ TITLES THEN ON ERR RESET:
PRINT #4: LOAD T$(VAL R$)
 110 GO TO 10
1000 ON ERR RESET
1010 ON ERR GO TO 1000
1020 CLS
1030 PRINT AT 10,4;"DON'T STOP THE PROGRAM!"
1040 FOR I=1 TO 10: BEEP .1,.01: BEEP .05,10: NEXT I
1050 CLS
1060 GO TO 10
2000 ON ERR CONTINUE
```

```
9000 LET TITLES = 8
9010 DIM T$(TITLES,10)
9020 FOR I=1 TO TITLES: READ T$(I): NEXT I
9030 CLS
9040 ON ERR GOTO 1000: RETURN
9900 DATA "PHONEBOOK"
9901 DATA "PANTRY"
9902 DATA "BIO-RHYTHM"
9903 DATA "FAMILY BIO"
9904 DATA "LOANER"
9905 DATA "TASWORD"
9906 DATA "MAIL LIST"
9907 DATA "BANNERS!"
```

Line 2 is the line that demonstrates how to use a printer driver that has been modified to AUTO LOAD. The name is changed to "p" from "AUTO" and can be called as done here. The titles in lines 9900 on should be changed to your own titles. I use this for our "family" disk at home. This can be safely RUN because there is a provision for STOPping it and DOS cannot load my filenames from your disk unless you have files with the same titles. Don't forget to change line 9000 to set TITLES to the number of files you have named in DATA lines.

As I have stated, this is a simple shell. A more elaborate one might include things like MOVEing from disk to disk, renaming, ERASEing, etc. Let's see what kind of ideas you can generate! As always, I am available for comments, questions, and problems at: 342 Trotter Court, Sanford, FL 32773.

ZEBRA/TIMEX FDD DISK SYSTEM

Directory Track Reader

Michael C. Finn

The following program was written on an old model Zebra Disk Drive. Owners of the newer FDD 3000 may wish to run this program and report on the results. I don't know whether there are any software differences between the two models.

The disk drive controller formats the 3 inch disk into 40 tracks, numbered 0 to 39, each containing 4 K bytes. Each track is subdivided into 16 sectors, numbered 0 to 15, each containing 1/4 K or 256 bytes. The first four tracks (0 to 3) are reserved for the operating system. Track 4 contains the directory. Tracks 5 through 39 contain the files you saved to disk. When a disk is intially formated, TOS sets up 40 tracks, writes a copy of TOS to tracks 0 through 3, places the disk name in the first sector of track 4 and sets all unused bytes to 0E5 h (229 d).

The Zebra Disk Drive Technical Manual remarks that the contents of the directory track can be read using the extended Basic command INPUT *#0. After some trial and error, I found a suitable method to read this track using a Basic program.

I numbered this program beginning with line 200. You may wish to merge this program with a utility program which reads disk headers. One such program is Chuck Dawson's DISKREAD, which was published in Vol., No.1 of the (now defunct) T.O.P.S. newsletter. DISKREAD, or a similar utility, could be used to obtain further details on the directory contents.

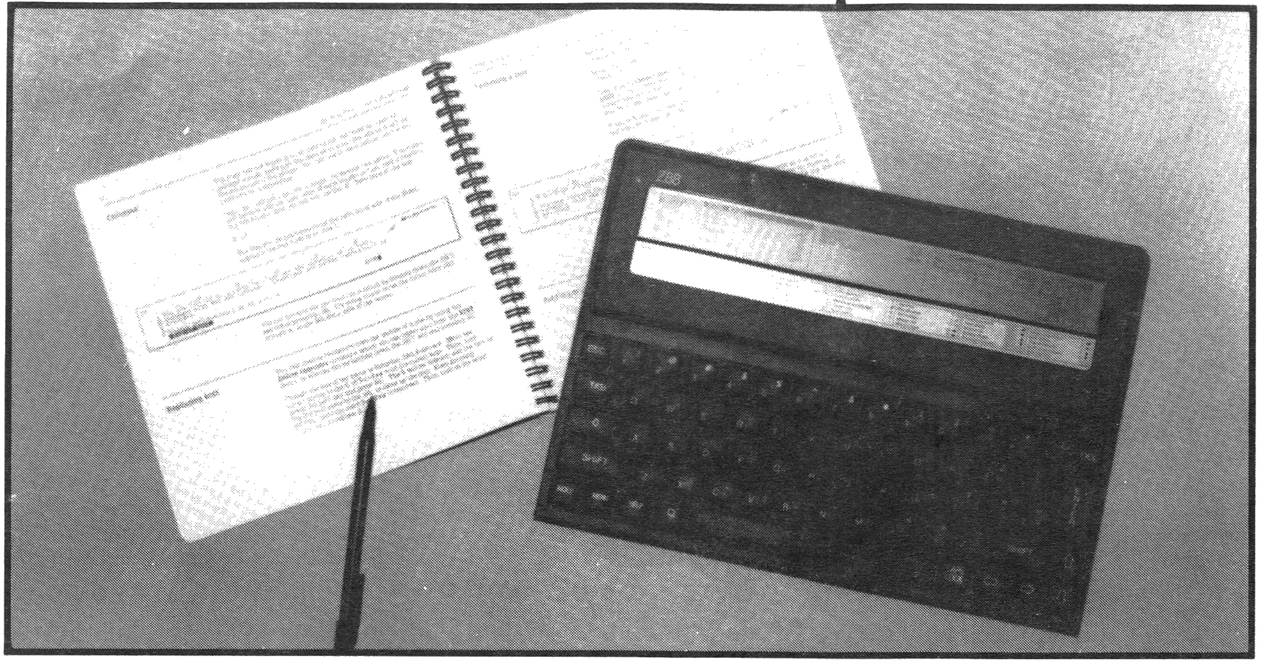
Lines 205 to 210: You have a choice of sending output to the screen or direct to your printer. The printer could be an 80 column printer. The 80 column printer drive I use, adjusts the addresses used for channel number 3, so that this channel drives my 80 column printer. For those without a full size printer, channel 3 contains the TS2040 I/O addresses from normal RAM initiation on initial power-up.

Line 215: The CAT * command is required to remove a "bug". After I first ran this program, I inserted a new disk and ran the program again. Instead of getting the directory of the new disk, I got the directory of the prior disk. Adding the CAT * command reset whatever internal buffer is involved with "#0". Note that 0 is not a valid channel number in other TOS commands.

[Editor: Space doesn't allow us to print all of the program notes for this listing. If you would like to receive a copy of the entire documentation, just send a legal SASE to TIME DESIGNS, and we will send you a copy]

```
200>REM DIRECTORY READER
       by Mike Finn
 205 PRINT AT 10,0;"SELECT PRINT
OUT DEVICE :"''''"1  TV SCREEN"''
"2  PRINTER"
 208 INPUT  LINE Z$: IF Z$<"1" O
R Z$>"2" THEN  GO TO 208
 210 LET Z=1+VAL Z$
 215 CLS : CAT *
 220 PRINT #Z'''"THIS PROGRAM WIL
L ITEMIZE THE   ENTRIES ON TRACK
 4 OF THE DISK."
 225 PRINT #Z
 230 INPUT "PRESS ENTER TO CONTI
NUE. "; LINE B$: CLS
 240 RESTORE *#0
 300 FOR I=1 TO 128
 310 INPUT *#0;A$;AT I
 320 IF CODE A$(1)=229 THEN  PRI
NT #Z;"RECORD ";I;" IS EMPTY.":
GO TO 720
 325 PRINT #Z'"RECORD: ";I
 330 PRINT #Z;"LEVEL: ";CODE A$(
1);" ";"DISK NAME" AND (CODE A$=
255)
 340 PRINT #Z;"NAME: ";A$(2 TO 9
)
 350 LET A=CODE A$(10): IF A>=12
8 THEN  LET A$(10)=CHR$ (A-128):
 PRINT #Z;TAB 6;"PROTECTED CATAL
OG ENTRY"
 360 LET A=CODE A$(11): IF A>=12
8 THEN  LET A$(11)=CHR$ (A-128):
 PRINT #Z;TAB 6;"INVISIBLE CATAL
OG ENTRY"
 370 PRINT #Z;"TYPE: ";A$(10 TO
12)
 380 IF A$(10 TO 12)="DIR" THEN
 PRINT #Z'"THIS DIRECTORY CONTAI
NS ALL      FILES AND SUBDIRECTOR
IES WITH    LEVEL NUMBER ";CODE A
$(17);".": GO TO 700
 390 PRINT #Z;"EXTENSION #: ";CO
DE A$(13)
 400 PRINT #Z;"BYTES LAST SECTOR
: ";CODE A$(14)
 410 LET S=(256*CODE A$(15)+CODE
 A$(16))/2
 420 PRINT #Z;"# SECTORS WITH DA
TA: ";S
 430 PRINT #Z;"BLOCK ALLOCATION:
"
 440 FOR J=17 TO 24
 450 PRINT #z;TAB 6;CODE A$(J);T
AB 16;CODE A$(J+8)
 460 NEXT J
 500 LET T=(S-1)*256+CODE A$(14)
+(256 AND (CODE a$(14)=0))
 510 PRINT #Z;"TOTAL FILE BYTES:
";T
 520 LET A1=INT (S/4)
 530 LET A2=S-4*A1
 590 PRINT #Z;"UNUSED BYTES IN A
LLOCATION: ";1024*(A1+(1 AND (A2
<>0)))-T
 600 LET A1=INT (CODE A$(17)/4)
 610 LET TRACK=4+A1
 620 LET A2=CODE A$(17)-4*A1
 630 IF A2=0 THEN  LET A2=4
 640 LET SECTOR=16-4*A2
 650 PRINT #Z;"ALLOCATION BEGINS
:"'TAB 6;"TRACK ";TRACK;TAB 16;"
SECTOR ";SECTOR
 700 INPUT "PRESS ENTER OR KEYWO
RD STOP "; LINE B$: IF B$=CHR$ 2
26 THEN  GO TO 800
 710 CLS
 720 NEXT I
 800 RESTORE *#0
 810 STOP
```

A FIRST LOOK
Sir Clive's Newest Microchip Wonder



Tim Woods

INTRODUCTION

Let's just imagine for a moment that we have access to a Time Machine (the stuff science fiction is made of). And while we're at it, let's set the dial backwards into time to say, mid 1982. Not a remarkable year, but if your still following along with this scenario, you may remember the significance of this time, especially if you are a confirmed Sinclair "junkie".

Yes, this was the golden age of the small home computer, and Sir Clive Sinclair (known fondly as "Uncle Clive" by his followers) introduced a new personal computer to the American public called the ZX81. It was actually a more powerful version of an earlier model (the ZX80).

The Sinclair ZX81 was small, light weight, and was black in color. It had an unusual keyboard. The microprocessor was a Z80 (over all, there were only about four chips internally) and the operating system was surprisingly powerful considering it was the first computer to sell for under $100.

Now let's power up our Time Machine again, and jump back to where we started. It's 1987 (almost 1988). Five full years later and Sir Clive has just introduced a new computer again. It's a battery-powered, self-contained, laptop computer called the Z88.

Remarkably, the Z88 possesses many similarities with its predecessor, the ZX81. It is small (8" X 11.5" and less than an inch thick), it weighs right at two pounds with the batteries installed, and it is black in color. And yes, keeping true to tradition, it has an unusual keyboard...more on that shortly. The microprocessor is a Z80 (internally there are only four chips), and the computer's operating system is extremely powerful for it's size and cost.

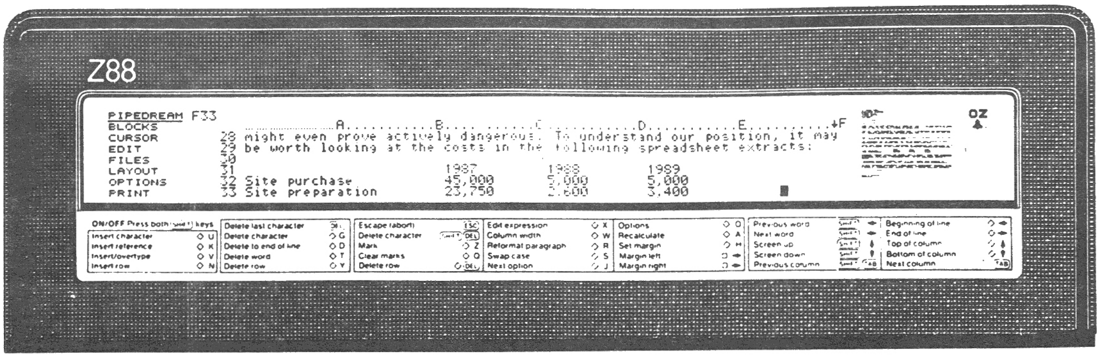
AN OVERVIEW

The Z88 computer tested for this review was purchased from Sharp's Inc. (Rt 10, Box 459, Mechanicsville, VA 23111, phone 804-746-1664) for $399.95. Mark Steuber of Sharp's is importing them from Great Britain. But currently, the Z88 sells for around £299 in England. But the U.S. dollar is extremely weak right now and with the current currency exchange rate, the computer would actually cost $530. The suggested retail cost for the U.S. model will reportedly be $499.

Another credit for help in this review goes to Mr. Sven Nilsson of the Las Vegas Timex Sinclair Users Group. He attended the Fall COMDEX and was able to get ahold of a copy of the official press packet (and also a photograph of Sir Clive showing off his new machine; unfortunately I was unable to get the photo ready for this issue...will print it next issue). The press packet related some information not included in the computer's User Guide.

Our computer arrived safely in the mail. It's packaging material has a plastic handle and doubles as a carrying case. Only the Z88 computer and the User Guide are in the box. Gone are the usual array of power supplys, cables, and TV switch box. The User Guide is very good..clearly written and informative. I believe someone who has never been exposed to a computer before would feel comfortable with the Z88. The manual that accompanies the Sinclair QL is awful compared to the simplicity of the Z88's User Guide. My only complaint is that there could have been more coverage in some areas, but I assume that additional documentation will be released soon.

Four AA alkaline batteries are required. They are inserted in a slot on the underneath side. The manual states that a fresh set of batteries will last 20 hours of heavy computer use (even with optional RAM cards installed). I used DURACELL's and was able to run the computer for about 30 hours before it automatically signalled that the batteries needed to be changed. When it is time to install new ones, an internal "super capacitor" keeps a charge to the RAM for up to six minutes, all data stored in memory will remain intact. Supposedly, you can store data for up to one year in the computer (this is with the power switch off) with a set of batteries. There is also a connection for a 6 volt power adapter, for when you use the Z88 at home. This connector is on the left side of the computer.



PIPEDREAM

Screen display example shows how PIPEDREAM can mix both word processor and spreadsheet functions to prepare a report. On the left side are the seven menu options. On the right side of the screen is an area called the "display map". It is a graphic representation of a sheet of paper. Since only eight lines are visible in the work space, the display map is a handy tool.

A plastic stand on the underneath side of the Z88 flips down to provide a nice tilt when using it on a table top. The keyboard is more comfortable this way, and the display is easier to see.

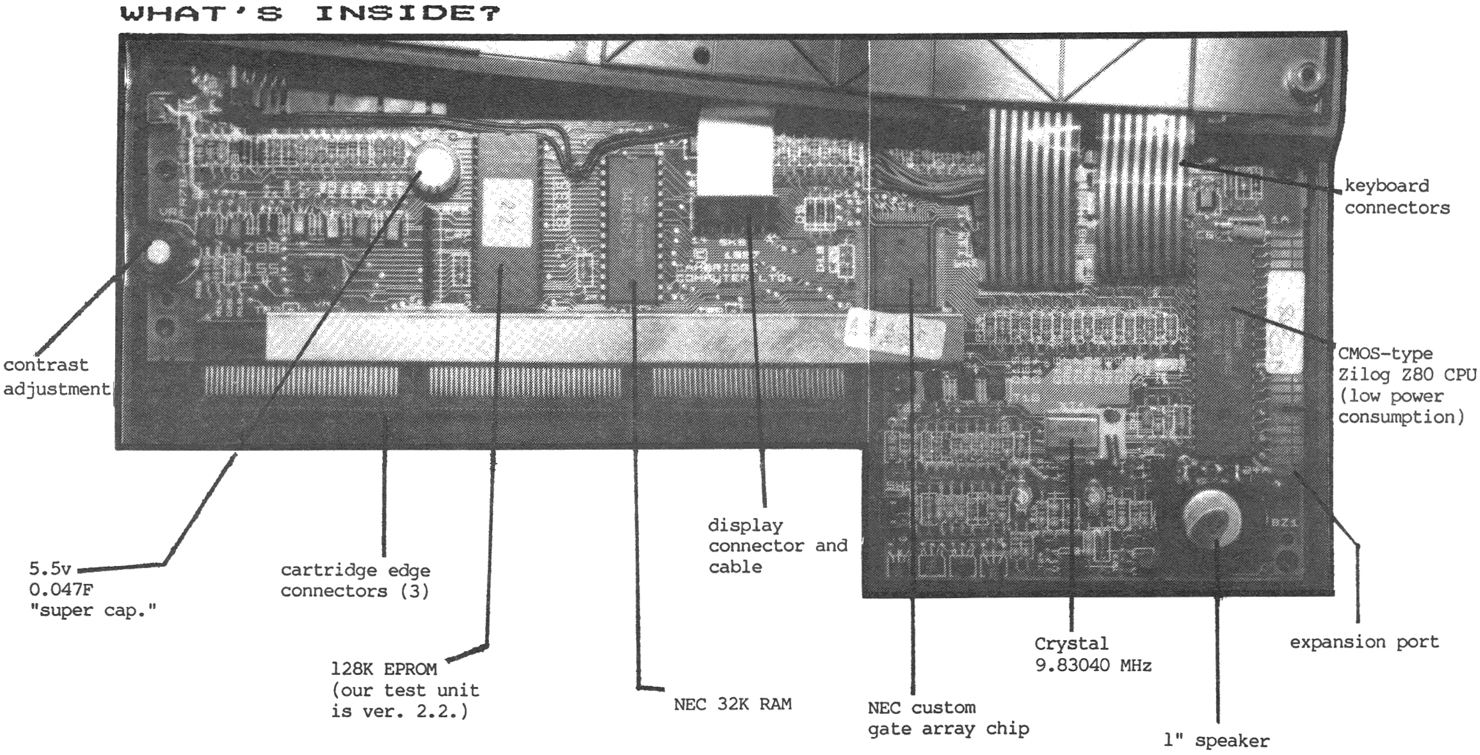
The built in LCD display is an interesting bit of high technology. It was designed and manufactured by Epson in Japan using the new Super-Twist display (name is derived by the way the liquid crystal molecules contort themselves to block the light whenever a voltage is applied). The screen packs a lot of information--8 lines by up to 90 characters wide. It is fast and there is no "flicker" as is usual for garden variety LCD displays. A small knob on the left side of the computer adjusts the constrast of the display for using the computer in different light source situations. However a brightly lighted room is ideal. Lower light levels tend to strain the eyes a bit. All of the screen characters and graphics are a nice blue color, while the background is grey.

My first impression of the keyboard was somewhat different than what it is now. The Z88 keyboard is interesting in that it is molded from a single sheet of soft latex rubber and is bonded to a standard membrane switchboard. Each key is raised, but travels just a small distance to make a contact. I guess the "feel" could best be described as that of a TS1500 (or early Spectrum) with the size and spacing of a TS2068. At first I was skeptical, but I quickly adapted to it, and now I actually like it (even for touch-typing). The intended purpose of this type of keyboard, according to Sir Clive, is to be silent in a classroom or even a library environment...it doesn't have the "clicking" sound of a full travel keyboard with real contacts. There is also another benefit that I discovered. Since it is just one piece, if something is spilled onto the keyboard, it could be easily wiped up, without permanently damaging key contacts.

The Z88 contains the simplest operating system I have ever used. Cambridge Computer (Sir Clive's new company responsible for the development of the Z88) calls the operating system "OZ" (like the clever wizard in the Frank Baum classic children's book). The casual or novice user simply moves about the screen with the cursor keys, a bar highlights commands on the menu, when a particular function or command is located (and highlighted with a bar) the user simply presses ENTER. The more advanced user can commit special commands to memory to bypass the menu-driven function. Using the special diamond ◇ and box □ command keys along with one or two other keys, all of the functions are quickly available in an instant. The most frequently used commands are permanently printed on a panel just below the Z88's display.

Continued Next Page.

WHAT'S INSIDE?



BUILT-IN SOFTWARE

When the Z88 is first powered-up, the user is confronted with a small menu box on the left side of the display. It shows what programs and utilities are on-board and permanently stored in the computer's memory. Internal software is divided into two types: "applications" and "popdowns".

"PIPEDREAM" is the most unique and most powerful application program included with the Z88. It is a combination of a fully implemented word processor, a spreadsheet, and a simple database.

Another application program is the "DIARY". It is like having a pad of paper to jot down notes or keep a journal (it can be controlled and used in coordination with the internal calendar, clock, and multi-function alarm clock).

The Z88 comes with BBC BASIC. Old Sinclair hands may turn up their nose upon hearing this. BBC BASIC is very close to Microsoft BASIC, and is used in several European computer brands. Since Sir Clive sold his own Sinclair BASIC and SuperBASIC to Amstrad, he had to select another version. Perhaps the saving grace of this BASIC is that it includes a built-in Z80 machine code assembler. Users can then by-pass the BASIC and program in code.

Other application programs include a terminal (emulates the VT52 protocol) for telecommunications and a built in printer driver which can be customized.

"Popdowns" are different in that they can be called up instantly from any point, even while working on any of the application program, and then exit back to the program with out interrupting anything you were working on. Real multi-tasking on a Z80 machine! An example of how this would work, would be--using the word processor for preparing a year-end financial report for your company, you may need to use the calculator to do some quick figuring--easy to do on the Z88.

"Popdowns" included with the Z88:

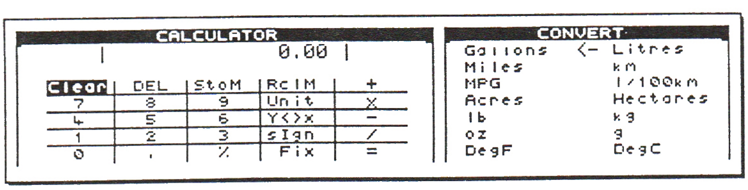
"INDEX" - Similar to a main menu from which other functions branch, but also keeps track of the status of the optional memory cards.

"CALCULATOR" - Includes scientific functions and a units-conversion facility.

"CALENDAR" - A perpetual calendar, and any month or year can be examined. Might be useful for students of history.

"CLOCK" - Displays current date and time.

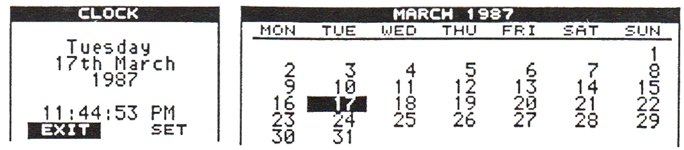
"ALARM" - A fully programmable multi-function alarm which can also display messages and reminders. Can even be set to execute and run a program.



"FILER" - When files have been created in any of the application programs, they can be sent to the Filer. From here you can store the data files in internal RAM, option RAM cards, on an EPROM card, or mass storage devices (disk drives) that may be developed for the Z88 in the future. Files can be stored in a simple directory or the option of using a flexible hierarchical directory system similar to MS-DOS.

"IMPORT/EXPORT" - This utility allows the user to transfer data and files via the built-in serial port to another external computer, and retrieve them back again. This function is just another way that the Z88 can save information. Special software packages (required) are being developed along with cable sets to attach and use virtually any popular computer. The software will signal the external computer to open a file. The only requirement is that the computer be equipped with a serial port. Available packages at this time include the IBM PC (and compatibles), the BBC Micro (European), and the Sinclair QL. Available soon for the Commodore 64/128, Atari ST, Apple II series, and the MacIntosh. The software package for the IBM is called "PC-LINK" and includes a utility on disk that converts Z88 files into WORDSTAR and LOTUS files.

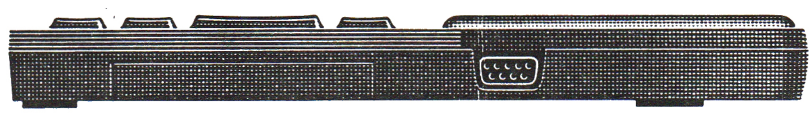
"PANEL" - Let's the user customize the Z88. Functions include: Auto key repeat rate, turn the speaker on/off, keyclick on/off, and setting parameters of the built-in RS232 Serial port (baud rate, parity, etc.).



THE PORTS

As mentioned above, there is a standard RS232 port built in, on the right side of the Z88. It can be used to attach the computer to a printer, or to a modem. The serial port is also the link to an external computer.

Also on the right side, is an expansion port. No pin-out diagram of this port was available in the literature I received. Several of the pins are direct unbuffered connections to the Z80 CPU. The User Guide does hint at a disk drive system that Cambridge may offer. Peripherals like the disk drive, and perhaps a board that will allow the Z88 to be connected to a standard monitor, will be attached to this port.



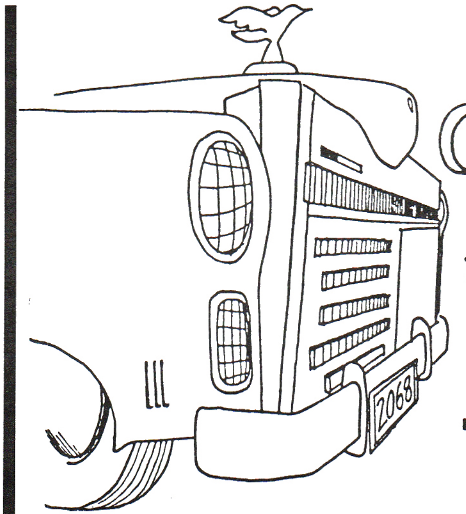
Just underneath the front lip of the Z88 is a hinged clear plastic door that reveals three cartridge docks. They are clearly labeled "1, 2, and 3". When the door is opened, the Z88 automatically powers down as a safety feature (when closed again, and power is restored, any work in progress will not be distrubed).

Several types of memory cards (or cartridges) are available from Cambridge Computer. Extension RAM cards can be used in all three slots. Currently, 32K and 128K RAM cards can be purchased. The Z88 has 32K built in (only about 20K is available for BASIC programming and data files). With three 128K cards installed, plus the internal RAM, one would have approximately 400K to work with. But the big news is that 1 Mbyte RAM cards will be available early next year. A total of 3 Mbyte with three of these cards on board! Cambridge states that the entire works of Shakespeare can be stored in RAM. This much power is almost unheard of in a battery powered laptop.

EPROM cards can also be used in all three slots (32K and 128K) for preprogrammed software or data files. A blank EPROM can be fitted into slot three, which has a built in EPROM "burner" circuit to permanently save data.

ONLY SURFACE SCRATCHED

We have just scratched the surface. This is an impressive little computer and we will continue this discussion next issue. All manner of possibilities that involve our other beloved Sinclairs can be imagined (how about carrying your TS2068 or QL files with you wherever you go?). I have demonstrated this computer to quite a number of people...everyone I show it to, wants one! A few balk at the price. If Sir Clive ever gets the price down below the $399 - $499 range, the whole world will want one of these.



CLASSY FRONT END

PART FOUR

BY PAUL BINGHAM

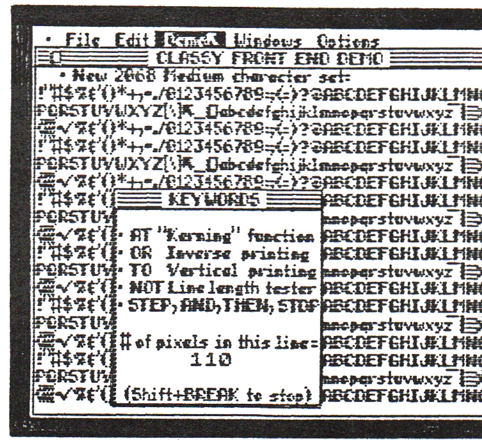


Figure A

Greetings programmers! As promised we will be installing the rest of the code, completing our machine language version of CLASSY! Some anxious users have sent for advanced copies of CLASSY and many others have written with suggestions and comments. We will look at these as well. (And thanks again for all your letters!)

First let us review what we have done up until now:

-In PART ONE (Mar/Apr '87) was the stand-alone BASIC CFE.

-In PART TWO (Jul/Aug '87) we discussed the design of a machine code version and all the font code was given as DATA statements (comprising lines 1000 thru 1190).

-In PART THREE (Sep/Oct '87) The beginning code was given (comprising lines 1 thru 90, the rest being a demo).

-In this portion (PART FOUR) we will add the last code and try out all of its features!

Before we add the new code, there are a change or two we need to make to the code from parts Two and Three. The needed alterations are shown in Figure B (and in Listing 3's first small block if you are using an assembler). The +350 in line 7 must also now become +1501. Some refinements to the letter "N" and chr$ 199 are given in Figure C, but these can be left out if you wish.

If you have lines 1000 thru 1190 of Part Three and have made the changes as noted, we can add the final code! This is Listing 1, lines 100 thru 220. (A disassembly of this code appears as Listing 3). When this has been entered our CLASSY coded version is complete. It contains all of the code to function as well as all of the new font data and the BASIC version will load itself when RUN.

But in order to see CLASSY in action, we need to give it something to do! Listing 2 is a demo which produces the dummy menu screen shown in Figure A. If Listing 2 is too long for you just type in lines 2000 thru 2100 and it will produce the same screen (minus the window). Once this is in place (and SAVEd on tape) RUN the program and enjoy. The initial POKE loop to load the 1501 bytes of code takes 18 seconds. After a first run you can redraw th screen with a simple GO TO 2000 (no need to re-RUN and wait eighteen seconds to re-load code already in place.

Let's look at the demo (Listing 2) in detail to better see the features CLASSY has built into it and how to work with them. A quick glance at the listing will show many lines with REM statements and "RANDOMIZE USER print". The REM statement contain the PRINT coordinates and the items for CLASSY to print. The USER calls are always in the line just previous to a REM line and call CLASSY into action. There can be many other commands in the same line with the user call (see line 2005) as long as the very next thing is a new REM line. If not you will get an ERROR A.

The demo screen (Figure A) shows the complete CLASSY font as produced by lines 2050, 2070, and 2090. Most of the symbols are similar in the CLASSY font with the few exceptions as discussed in PART TWO. You will notice in line 2050 two apostrophes after the "!" in that line. This is a departure from Sinclair BASIC (Clive, forgive me!). In CLASSY two apostrophes = one quote symbol.

In lines 2070 and 2090 you will notice several ATs. AT is the kerning keyword to fit letters (like "A" and "T") closer together. Each AT equals one pixel backspace. Line 2270 is replete with them. Another keyword (line 2010) is OR. This toggles in and out of inverse mode. Note that the word DEMO and arrow are printed inversely. Now since CLASSY printing is always in OVER mode, OR can also be used with INK and PAPER to erase words or we can use the old stand-by: PRINT "(spaces) ".

The keyword TO appears in line 2250 and toggles the vertical print mode. TO is real handy for columns! Notice that this line prints the dots downward in the demo's window area. Not the last keyword you'll find is NOT. NOT is the "Number of test" keyword. Since each CLASSY symbol has its own width, it would be nearly impossible to calculate a line's length without NOT. With NOT once in line (see line 2290) the sum (up to 255) is tucked away in the 38th byte of CLASSY's code. It needs only to be PEEKed (see lines 2300). But repeated PEEKing can be a chore. Using NOT twice in a line (see line 2340) will halt the program with ERROR = n. Looking up the chr$ code for "n" reveals it equals 110. And in this case 110 is the exact length of the line. NOT calculates all symbols, spaces, and ATs.

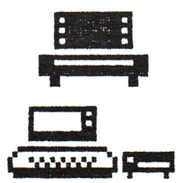
In order to get out of an ERROR loop just hit SHIFT+BREAK. The other ERROR messages are ERROR A for an improper word after REM, ERROR B for improper coordinate or variable, and ERROR C for an unknown symbol in the quotes. It's conceivable you could crash CLASSY but with these error traps it's pretty unlikely. Notwithstanding, it's always wise to "Save it" 'fore ya run it!"

The code also sorts for four more keywords which I've left user-definable. These are STEP, AND, THEN, and STOP. The jump table for these is eight bytes long and starts at the 530th byte in the code (at B1D9 in Listing 3). These are set now to jump to ERROR C, but you could reset these to your own custom routines. Where to put custom routines? Pretty simple, really. The font data sits on CLASSY like a cap. If we move the font data up we can gain the room we need. Just put the new offset number to the data into two pairs of bytes starting with the 507th byte (B1C2, Listing 3) and the 700th byte (B283, Listing 3). Then from the 711th byte (B28h) onward is room for new code!. There are also eight empty storage or flag bytes near the code's beginning starting at the 30th byte (AFE5, Listing 3).

Lemke Software Development
FALL 1987

Quality products for the TS 2068

DESKTOP PUBLISHING: THE ABILITY TO PRODUCE NEWSLETTERS AND OTHER DOCUMENTS OF TEXT AND GRAPHICS HAS JUST BEEN MADE AVAILABLE TO TS 2068 OWNERS...



PIXEL PRINT NEW PRINT SENSATION!

Lemke Software Development of Wichita, Ks. has just added the PIXEL PRINT Desktop Publisher to their program line!

This ad created with Pixel Print.

+ Single or Twin Columns
+ Variable Size Characters
+ Many Fonts + LOAD CUSTOM Fonts
+ SCREEN$ Graphics or ICONS
+ Full Size or TS 2040 printer (IBM compatible DOT Matrix)
+ COPY/ERASE/INSERT/DELETE/UNDO

only $19.95 ppd.

ICON PACKAGE

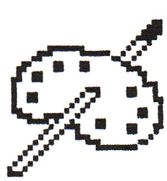
for the Desktop Publisher

+ 102 ICON Clip Art Pictures
+ ICON Librarian(browse ICONS)
+ ICON Designer

only $19.95 ppd.

SOON: ICON #2 (Fall 1987)
+ 100 more ICONS
+ $14.95 ppd. (no programs)

--ICONS BY MOUNTAINEER SOFTWARE---



PIXEL SKETCH AND GRAPHICS EDITOR

+ 32 Column Draw/Edit
+ Extended Color Mode Draw/Edit
+ 64 Column Mode (Hi Resolution)
+ Merge Screens
+ Full Screen (Window) Edit
+ ENLARGE/Shrink/Rotate/Mirror
+ TEXT (uses FONT PACKAGE too!)
+ Draw/Erase/Dash
+ IBM Dot Matrix Full Size and TS 2040 Printers
+ UNDO (oops!) function
+ Digitize (display bytes)

only $19.95 ppd.

FONT PACKAGE

for the Desktop Publisher

+ 6 New Fonts

HEADLINE	Adventure
IBM	Outline
MICR	5X5

+ Font Designer and Librarian

only $19.95 ppd.

— Font Package #2 —

+ 6 More Fonts

Helvetica	Ritz
Times	Sinclair
Blox Block	GrandPrix

only $14.95 ppd.
(no programs!)

--FONTS BY MOUNTAINEER SOFTWARE--

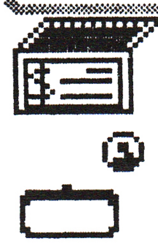
— Font Package #3 —

+ 6 More Fonts

Cursive	Thespian
Memento	Western
STENCIL	Wideload

+ 26 Graphic Borders
+ 12 Masthead (Banner) Designs

only $19.95 ppd.
(no programs!)

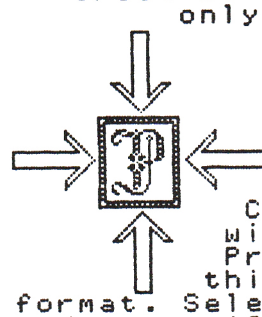


Checkbook Budget Master

Checkbook Database and Budget Analysis Program

+ 800 Check Database
+ Add/Delete/Recall Checks Fast
+ Categorize Expenses
+ Compare Income vs Expense
+ Great at Tax Time!

only $19.95 ppd.



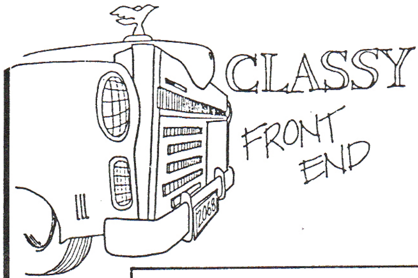
Pixel Print TASWORD Utility

Create your TEXT file with your TASWORD Word Processor and convert this to the Pixel Print format. Select any Font... locate and merge ICONS too!

only $19.95 ppd.

Lemke Software Development

S D Lemke
2144 White Oak
Wichita, Ks.
67207



CONTINUED FROM PAGE 15

TS2068

```
  10 DATA 0,0,0,0,0,0,0,0,0,0,0,
0,0,213,197,229,245,217,8,213,19
7,229,245,237,91,176,92,24,13,0,
0,0,0
  20 DATA 0,0,0,0,0,225,35,24,7,
42,65,92,1,5,0,9,126,254,173,32,
6,235,54,2,235,24,77,254,172,32,
6,235,54,1,235,24,67

1000 DATA 4,0,0,0,0,0,0,0.,3,0,9
4,94,0,0,0,0.,5,0,7,0,0,7,0,0.,6
,34,255,34,34,255,34,0.,6,0,44,1
10,255,74,48,0
```

Figure B

```
1090 DATA 7,0,126,126,4,8,4,126.
,7,0,126,126,12,48,126,126.,6,0,
60,126,66,66,60,0.,6,0,126,126,1
8,18,12,0.,6,0,60,126,62,98,252,
0

1190 DATA 5,0,24,60,231,36,0,0.,
6,255,119,219,119,219,255,0.,3,0
,24,24,0,0,0,0.,7,24,48,96,48,8,
4,2
```

Figure C

HEART OF CFE "PRINTING" ROUTINE (m.c. version)

```
         EXX                    swap 3 register-pair's contents
         POP BC                 let C=X value, B=Y value
         EXX                    swap contents back
         LD B,A                 put width in B
SYMBOL:  INC HL                 point to next graphic byte
         XOR A                  zero carry flag
         LD A, (HL)             put next graphic byte in A
         PUSH BC                save width
         LD B, 8                set counter to eight (8 bits!)
NEXTBIT: RLA                    A's left-most bit into carry
         EXX                    swap contents
         JR NC,Y UP             if carry=0 jump to Y UP
        (in inverse plotting this becomes JR C,Y UP)
         PUSH BC                save X,Y values
         PUSH AF                save graphic byte & carry flag
         CALL 9790 (263Eh)      call ROM plot routine
         POP AF                 restore AF
         POP BC                 restore BC
Y UP:    INC B                  increase Y value by one
         EXX                    swap contents back
         DJNZ, NEXTBIT          if 8 bits not done, to NEXTBIT
         POP BC                 width restored
         EXX                    swap contents (BC=X,Y values)
         INC C                  add one to X
         LD A, B                drop Y value by eight
         ADD A, 248 (F8h)
         LD B, A
         EXX                    swap contents back
         DJNZ, SYMBOL           if all bytes not seen to SYMBOL
```

Figure D

Now a couple of other tips before we move on. The REM lines 2030 and 2050 show that REM acts as PRINT does in a BASIC print statement. AT and the absence of coordinates is accepted as in Sinclair BASIC. TAB is also accepted. TAB can act even as a backspace. A line like REM TAB 10;"B";TAB 9;"3" will print 38!. If you use variables (a thru h) the values must first be POKEd into place. This was shown in the demo in PART III. INK and PAPER can be changed but must be done outside a REM line. Such commands in quotes will give ERROR C.

In the interest of space and time ('sounds like Carl Sagan!), I can't possibly detail every machine code routine in CLASSY, so I'll review the heart of CLASSY's crazy printing abilities. This 35 byte code is found in line 180, also at B1E9 in Listing 3, and is also shown in Figure D. We enter this routine with the following: The top stack values are the X and Y plot values, HL has the address of the first of eight bytes of font data for a particular symbol, and A contains the contents of that first byte (the width). If you are unfamiliar with Z80 code, I would recommend studying Syd Wyncoop's eleven lessons on Z80 code starting with the March/April '86 issue of Time Designs to present (all eleven issues are still in stock for $3 each). Specifically if you are not familiar with XOR, RLA, or EXX read lessons #6 and #7 (in Jan/Feb '87 and Mar/Apr '87). Lessons #4 and #5 contain almost all the other machine language instructions found in this print routine. Study Figure D and the remarks for each instruction and you will see how it was relatively easy to plot pixels vertically to form characters instead of horizontally like the ROM does it. This routine calls a 114 byte ROM plot routine starting at 9790 (263Eh) for each pixel plotted. The ROM routine actually starts at 9784 (2638h) but we by-pass 36 bytes of un-needed temp color code by entering late. In either case BC must contain the plot coordinates before the call. The order is C = X and B = Y.

I hope no matter what you level of skill in programming, that you will enjoy CLASSY FRONT END. It is truly fun to play around with. And you must play around with it to become familiar with it. If you have any questions or comments, please write. Now you may save yourself some embarrasment (and me some paper) if you read PART II, III, and IV carefully first...many of the recent questions I answered by referring to a past article. Nonetheless, if you have a problem, write. No question is a bad one if your learning! Next time perhaps we will tackle true windowing in conjunction with our new program. Until then, may all your drives be snappy--and all your monitors be bright! And HAPPY HOLIDAYS!

***If you would like a copy of the complete machine code version of CLASSY FRONT END on a new cassette, just send $5.00. I also offer a cassette of many past programs called "BINGHAM's BEST" which also contains the BASIC and the machine code versions of CLASSY all for $9.95. Just send it to: Paul Bingham, PO Box 2034, Mesa, AZ 85214.

Listing 1

```
 100 DATA 35,24,2,24,141,126,254
,34,32,51,35,126,254,58,32,18,22
9,33,38,0,229,46,11,124,25,119,3
5,126,214,8,119,225,25,233,254,5
9,40,5
 110 DATA 33,126,0,24,233,35,126
,254,13,32,5,33,126,0,25,233,43,
229,33,38,0,25,233,254,39,32,14,
35,126,254,39,32,4,62,34,24,8,43
,126,24,4
 120 DATA 203,127,32,20,6,7,229,
213,22,0,95,213,225,25,16,253,20
9,229,24,122,198,185,24,236,254,
199,40,248,254,200,40,244,254,20
1,40,240
 130 DATA 24,5,235,22,67,24,136,
254,197,40,77,254,195,40,114,254
,172,40,120,254,204,40,18,254,20
3,40,104,254,198,40,102,254,205,
40,100,254,226,40,98
 140 DATA 24,217,229,213,225,35,
126,254,1,40,7,54,1,1,138,24,24,
5,54,0,1,217,229,33,73,2,25,112,
35,113,24,123,229,8,217,197,245,
121,193,144
 150 DATA 193,79,120,198,248,71,
8,24,101,229,33,45,2,25,126,254,
56,40,4,54,56,24,2,54,48,24,88
 160 DATA 193,33,198,1,25,9,8,12
6,8,126,213,229,33,11,0,25,78,35
,70,225,197,24,18
 170 DATA 24,68,24,162,24,160,24
,158,24,156,229,33,11,0,25,53,24
,47
```

```
 180 DATA 217,193,217,71,35,175,
126,197,6,8,23,217,48,7,197,245,
205,62,38,241,193,4,217,16,241,1
93,217,12,120,198,248,71,217,16,
225
 190 DATA 209,33,11,0,25,229,217
,225,113,35,112,217,33,26,1,25,2
33
 200 DATA 62,0,50,119,92,35,126,
254,34,40,37,254,195,32,46,213,2
25,1,116,0,9,54,61,35,35,35,58,1
19,92,6,2,79,175,121,23,56,4,16,
251,14,32,121,119,33,92,0,25,233
 210 DATA 33,37,0,25,58,119,92,1
19,33,126,0,25,233,203,127,40,21
,1,199,2,185,40,13,12,16,250,254
,172,32,183,58,119,92,61,24,174,
198,185
 220 DATA 229,213,6,7,22,0,95,21
3,225,25,16,253,209,229,193,33,1
98,1,25,9,58,119,92,134,225,24,1
45
```

Listing 2

```
2000 REM     DRAW MAIN SCREEN

2005 PAPER 5: BORDER 1: CLS : PA
PER 7: FOR t=0 TO 7: PRINT "
  ";: NEXT t: PLOT 2,2: DRAW 2
50,0: DRAW 0,166: DRAW -250,0: D
RAW 0,-166: PLOT 2,159: DRAW 250
,0: RANDOMIZE USR print
2010 REM AT 0,1;">=  File  Edit
OR  Demo↑ OR  Windows  Options";
AT 1,0;" }£}}}}}}}} CLASSY FRONT
 END DEMO }}}}}}}}"
2020 PAPER 5: RANDOMIZE USR prin
t
2030 REM AT 2,2;">= New 2068 Med
ium character set:"
2040 FOR t=1 TO 6: RANDOMIZE USR
 print
2050 REM " !''#$%&'()*+,-./01234
56789:;<=>?@ABCDEFGHIJKLMNO"
2060 RANDOMIZE USR print
2070 REM "AT  PQRSTUVWXYZ[\]↑_£a
bcdefghijklmnopqrstuvwxyzAT AT {
 STICK } FREE "
2080 RANDOMIZE USR print
2090 REM " AT ©<=>=<>%&'()*+,-./
0123456789:;<=>?@ABCDEFGHIJKLMNO
"

2100 NEXT t
2190 REM       Keyword box

2200 PAPER 7: FOR t=9 TO 20: PRI
NT AT t,6;"                  ": NEXT
 t: PLOT 46,104: DRAW 0,-98: DRA
W 113,0: DRAW 0,1: DRAW -112,0:
DRAW 0,88: DRAW 112,0: PLOT 46,1
04: DRAW 114,0: DRAW 0,-97: PAPE
R 3: RANDOMIZE USR print
2210 REM AT 9,6;"}}}} KEYWORDS }
}}}}"
2220 PAPER 7: RANDOMIZE USR prin
t
2230 REM AT 11,7;"AT AT AT ''Ker
ning'' function ";AT 12,7;"AT OR
  Inverse AT AT  printing  ";AT
13,7;"AT TO  Vertical AT AT  pri
nting ";AT 14,7;"AT NOT AT AT Li
ne AT AT length AT tester ";AT 1
5,7;"AT AT STEP, AT AT AND,AT AT
 AT  THEN, AT AT STAT OP"
2235 REM       (Use of TO)

2240 RANDOMIZE USR print
2250 REM AT 11,6;" TO >=>=>=>= T
O >="
2255 REM       (Use of NOT)

2260 RANDOMIZE USR print
2270 REM AT 17,6;" AT AT # AT AT
 of AT pixels in AT this AT line
 AT AT ="
2280 RANDOMIZE USR print
2290 REM AT 17,6;"NOT  AT AT # A
T AT of AT pixels in AT this AT
line AT AT ="
2300 PRINT AT 18,11;PEEK 45037
2310 RANDOMIZE USR print
2320 REM AT 20,6;" (Shift+BREAK
to stop)"
2330 RANDOMIZE USR print
2340 REM AT 17,6;"NOT  AT AT # A
T AT of AT pixels in AT this AT
line AT AT =NOT "
```

Listing 3

```
AFDF ED5BB05C   LD DE,(nmia)
AFE3 180D       JR AFF2
AFE5 00         NOP
AFE6 00         NOP
AFE7 00         NOP
AFE8 00         NOP
AFE9 00         NOP
AFEA 00         NOP
AFEB 00         NOP
AFEC 00         NOP
AFED 00         NOP
AFEE E1         POP HL
AFEF 23         INC HL
AFF0 1807       JR AFF9

B0E3 23         INC HL
B0E4 1802       JR B0E8
B0E6 188D       JR B075
B0E8 7E         LD A,(HL)
B0E9 FE22       CP 22
B0EB 2033       JR NZ,B120
B0ED 23         INC HL
B0EE 7E         LD A,(HL)
B0EF FE3A       CP 3A
B0F1 2012       JR NZ,B105
B0F3 E5         PUSH HL
B0F4 212600     LD HL,0026
B0F7 E5         PUSH HL
B0F8 2E0B       LD L,0B
B0FA 7C         LD A,H
B0FB 19         ADD HL,DE
B0FC 77         LD (HL),A
B0FD 23         INC HL
B0FE 7E         LD A,(HL)
B0FF D608       SUB 08
B101 77         LD (HL),A
B102 E1         POP HL
B103 19         ADD HL,DE
B104 E9         JP (HL)
B105 FE3B       CP 3B
B107 2805       JR Z,B10E
B109 217E00     LD HL,007E
B10C 18E9       JR B0F7
B10E 23         INC HL
B10F 7E         LD A,(HL)
B110 FE0D       CP 0D
B112 2005       JR NZ,B119
B114 217E00     LD HL,007E
B117 19         ADD HL,DE
B118 E9         JP (HL)
B119 2B         DEC HL
B11A E5         PUSH HL
B11B 212600     LD HL,0026
B11E 19         ADD HL,DE
B11F E9         JP (HL)
B120 FE27       CP 27
B122 200E       JR NZ,B132
B124 23         INC HL
B125 7E         LD A,(HL)
B126 FE27       CP 27
B128 2004       JR NZ,B12E
B12A 3E22       LD A,22
B12C 1808       JR B136
B12E 2B         DEC HL
B12F 7E         LD A,(HL)
B130 1804       JR B136
B132 CB7F       BIT 7,A
B134 2014       JR NZ,B14A
B136 0607       LD B,07
B138 E5         PUSH HL
B139 D5         PUSH DE
B13A 1600       LD D,00
B13C 5F         LD E,A
B13D D5         PUSH DE
B13E E1         POP HL
B13F 19         ADD HL,DE
B140 10FD       DJNZ,B13F
B142 D1         POP DE
B143 E5         PUSH HL
B144 187A       JR B1C0
B146 C6B9       ADD A,B9
B148 18EC       JR B136
B14A FEC7       CP C7
B14C 28F8       JR Z,B146
B14E FEC8       CP C8
B150 28F4       JR Z,B146
B152 FEC9       CP C9
B154 28F0       JR Z,B146
B156 1805       JR B15D
B158 EB         EX DE,HL
B159 1643       LD D,43
B15B 1888       JR B0E5
B15D FEC5       CP C5
B15F 284D       JR Z,B1AE
B161 FEC3       CP C3
B163 2872       JR Z,B1D7
B165 FEAC       CP AC
B167 2878       JR Z,B1E1
B169 FECC       CP CC
B16B 2812       JR Z,B17F
B16D FECB       CP CB
B16F 2868       JR Z,B1D9
B171 FEC6       CP C6
B173 2866       JR Z,B1DB
B175 FECD       CP CD
B177 2864       JR Z,B1DD
B179 FEE2       CP E2
B17B 2862       JR Z,B1DF
B17D 18D9       JR B158
B17F E5         PUSH HL
B180 D5         PUSH DE
B181 E1         POP HL
B182 23         INC HL
B183 7E         LD A,(HL)
B184 FE01       CP 01
B186 2807       JR Z,B18F
B188 3601       LD (HL),01
B18A 018A18     LD BC,188A
B18D 1805       JR B194
B18F 3600       LD (HL),00
B191 01D9E5     LD BC,E5D9
B194 214902     LD HL,0249
B197 19         ADD HL,DE
B198 70         LD (HL),B
B199 23         INC HL
B19A 71         LD (HL),C
B19B 187B       JR B218
B19D E5         PUSH HL
B19E 08         EX AF,AF'
B19F D9         EXX
B1A0 C5         PUSH BC
B1A1 F5         PUSH AF
B1A2 79         LD A,C
B1A3 C1         POP BC
B1A4 90         SUB B
B1A5 C1         POP BC
B1A6 4F         LD C,A
B1A7 78         LD A,B
B1A8 C6F8       ADD A,F8
B1AA 47         LD B,A
B1AB 08         EX AF,AF'
B1AC 1865       JR B213
B1AE E5         PUSH HL
B1AF 212D02     LD HL,022D
B1B2 19         ADD HL,DE
B1B3 7E         LD A,(HL)
B1B4 FE38       CP 38
B1B6 2804       JR Z,B1BC
B1B8 3638       LD (HL),38
B1BA 1802       JR B1BE
B1BC 3630       LD (HL),30
B1BE 1858       JR B218

B1C0 C1         POP BC
B1C1 21C601     LD HL,01C6
B1C4 19         ADD HL,DE
B1C5 09         ADD HL,BC
B1C6 08         EX AF,AF'
B1C7 7E         LD A,(HL)
B1C8 08         EX AF,AF'
B1C9 7E         LD A,(HL)
B1CA D5         PUSH DE
B1CB E5         PUSH HL
B1CC 210B00     LD HL,000B
B1CF 19         ADD HL,DE
B1D0 4E         LD C,(HL)
B1D1 23         INC HL
B1D2 46         LD B,(HL)
B1D3 E1         POP HL
B1D4 C5         PUSH BC
B1D5 1812       JR B1E9
B1D7 1844       JR B21D
B1D9 18A2       JR B17D
B1DB 18A0       JR B17D
B1DD 189E       JR B17D
B1DF 189C       JR B17D
B1E1 E5         PUSH HL
B1E2 210B00     LD HL,000B
B1E5 19         ADD HL,DE
B1E6 35         DEC (HL)
B1E7 182F       JR B218
B1E9 D9         EXX
B1EA C1         POP BC
B1EB D9         EXX
B1EC 47         LD B,A
B1ED 23         INC HL
B1EE AF         XOR A
B1EF 7E         LD A,(HL)
B1F0 C5         PUSH BC
B1F1 0608       LD B,08
B1F3 17         RLA
B1F4 D9         EXX
B1F5 3007       JR NC,B1FE
B1F7 C5         PUSH BC
B1F8 F5         PUSH AF
B1F9 CD3E26     CALL 263E
B1FC F1         POP AF
B1FD C1         POP BC
B1FE 04         INC B
B1FF D9         EXX
B200 10F1       DJNZ,B1F3
B202 C1         POP BC
B203 D9         EXX
B204 0C         INC C
B205 78         LD A,B
B206 C6F8       ADD A,F8
B208 47         LD B,A
B209 D9         EXX
B20A 10E1       DJNZ,B1ED
B20C D1         POP DE
B20D 210B00     LD HL,000B
B210 19         ADD HL,DE
B211 E5         PUSH HL
B212 D9         EXX
B213 E1         POP HL
B214 71         LD (HL),C
B215 23         INC HL
B216 70         LD (HL),B
B217 D9         EXX
B218 211A01     LD HL,011A
B21B 19         ADD HL,DE
B21C E9         JP (HL)
B21D 3E00       LD A,00
B21F 32775C     LD (5C77),A
B222 23         INC HL
B223 7E         LD A,(HL)
B224 FE22       CP 22
B226 2825       JR Z,B24D
B228 FEC3       CP C3
B22A 202E       JR NZ,B25A
B22C D5         PUSH DE
B22D E1         POP HL
B22E 017400     LD BC,0074
B231 09         ADD HL,BC
B232 363D       LD (HL),3D
B234 23         INC HL

B235 23         INC HL
B236 23         INC HL
B237 3A775C     LD A,(5C77)
B23A 0602       LD B,02
B23C 4F         LD C,A
B23D AF         XOR A
B23E 79         LD A,C.
B23F 17         RLA
B240 3804       JR C,B246
B242 10FB       DJNZ,B23F
B244 0E20       LD C,20
B246 79         LD A,C
B247 77         LD (HL),A
B248 215C00     LD HL,005C
B24B 19         ADD HL,DE
B24C E9         JP (HL)
B24D 212500     LD HL,0025
B250 19         ADD HL,DE
B251 3A775C     LD A,(5C77)
B254 77         LD (HL),A
B255 217E00     LD HL,007E
B258 19         ADD HL,DE
B259 E9         JP (HL)
```

YOUR RAM MEMORY

EARL V DUNNINGTON

Here are some corrections to the first installment of this series which appeared in the July/August '87 issue of TDM. On page 30, lines 5 and 6, right column, should read as follows: "displacement between E_LINE and STKEND as after the line has been placed in the Program area, the Line being typed + work space area no longer exists and the".

Hopefully no one with a ZX81 that did not have a RAMPACK attached tried to do the examples from lines 27 to 30, left col., page 30. If so, I extend my apologies as there is not enough memory left after expanding the Display file to enter the commands. You would have to PEEK each address of the System variable E_LINE separately, recording the results either with pencil and paper or use the TS2040 printer and LPRINT instead of PRINT. Then turn off the computer to get back into the unexpanded Display mode. You can then carry out the arithmetic. The remainder of the examples can be done in the unexpanded Display mode.

In the first episode of this article, a diagram (Figure 1) was presented depicting the structure of the BASIC system in the RAM memory after turning on the computer or entering NEW. You will need that diagram, the TS2040 printer, and your USER MANUAL. BASIC routines were presented to prove that the structure of the BASIC system of the lower RAM memory consisted of either 153 or 920 bytes depending upon the value stored in the System variable RAMTOP. Partial proof was also presented that the number of bytes in the upper RAM memory used by the BASIC system after initialization, are four. Unfortunately, there is no System variable for the Machine stack pointer, sp. These two registers (s and p) of the Z80 CPU cannot be PEEKed directly using BASIC. Only by the use of a machine code program and the USR function can the address stored in these two registers be determined. Indirect proof that the addressed stored in the s and p registers and the address stored in the System variable ERR SP are the same at the time of completion of initialization, is that the routines I presented to set RAMTOP within a program would not work if this was not correct (see page 9 and 10, July/Aug '85, or page 9 and 10 of "The Best of TDM Vol.1", also Fig. No.1, pg. 16, Nov/Dec '86 and Fig. 4, pg. 40, Jan/Feb '87).

Referring to diagram Figure 1 (July/Aug '87, pg. 30), let us now prove that the System variables area contains 125 bytes. To do this we substract the fixed address of the start of the System variables from the fixed address 16509 which is the address of the first byte of the Display file in no BASIC program is in the computer. Turn on the computer and type in the direct command:

PRINT 16509-16384

Then press ENTER. The result should be 125.

The two bytes of the System variable D_FILE contain the address of the start of the Display file. If Figure No.1 is correct, PEEKing these two bytes should return the value 16509, proving that the Program area does not exist until a numbered program line is entered. Compare Figure No.1 with the diagram on page 128 of the ZX81 and the TS1000 (or page 154 of the TS1500) User Manuals. Turning to the listing of the System variables in your manual, you will find that the addresses of the two bytes of D_FILE are n=16396 and n+1=16397. Substituting these values in the PEEK formula at the beginning of the listing, type in the direct command:

PRINT PEEK 16396+256*PEEK 16397

Then press ENTER. The result should be 16509.

To prove that the unexpanded Display file, with the cursor showing on the screen, uses 26 bytes, we must be very devious, because when a direct command is ENTERed both the command and the cursor no longer occupy the Display file. In order to avoid using the screen for the results we use the 2040 or equivalent printer. Then if your computer has 3.25K or more RAM, you must fool the computer into thinking there is less than this amount available so that an unexpanded Display file will be set up. To do this, POKE the System variable RAMTOP with an address less than 19712 and then execute a CLS. For example using the address 19711, type in the following direct commands:

(Note: ZX81 must have RAMPACK)
POKE 16388,19711-256*INT (19711/256)
Then press ENTER
POKE 16389, INT (19711/256)
Then press ENTER
CLS
Then press ENTER

Everybody type in the program presented in Figure Number 2. A line by line explanation follows:

```
  10 POKE 16418,0
  20 PRINT AT 23,0;"K"
  30 LPRINT (PEEK 16400+256*PEEK
 16401)-(PEEK 16396+256*PEEK 163
97)
  40 LPRINT "PRESS ANY KEY TO QU
IT"
  50 LPRINT
  60 PAUSE 32768
  70 POKE 16418,2
```

FIGURE NO. 2

10 - POKEs the System Variable DF_SZ with a zero so that you can print on lines 22 or 23 of screen display.

20 - PRINTs the inverse K on line 23.

30 - LPRINTs, using the printer, the difference between the addresses contained in the System variables VARS and D FILE.

40 - LPRINTs a message on the printer.

50 - Advances the paper in the printer so that the messages can be read.

60 - Stops the program execution so that the screen with the inverse K can be seen.

70 - Restores the original value in the System variable DF_SZ to avoid a crash.

Turn on the printer, type RUN, and press ENTER. Be sure to follow the instructions to quit. Leave the program in your computer.

In order to prove that the expanded Display file, with or without the cursor, uses 793 bytes, POKE the System variable RAMTOP with a value 19712 or greater, then enter a CLS. For example, using 19712, type in the following direct commands:

POKE 16388, 19712-256*INT (19712/256)
Then press ENTER
POKE 16389, INT (19712/256)
Then press ENTER
CLS
Then press ENTER

Be sure that the printer is on, then type in RUN and press ENTER. To continue follow the instruction to quit.

Continued Next Page.

"CLASSY FRONT END"

(Listing 3 continued)

```
B25A CB7F        BIT 7,A
B25C 2815        JR Z,B273
B25E 01C702      LD BC,02C7
B261 B9          CP C
B262 280D        JR Z,B271
B264 0C          INC C
B265 10FA        DJNZ,B261
B267 FEAC        CP AC
B269 20B7        JR NZ,B222
B26B 3A775C      LD A,(5C77)
B26E 3D          DEC A
B26F 18AE        JR B21F
B271 C6B9        ADD A,B9
B273 E5          PUSH HL
B274 D5          PUSH DE
B275 0607        LD B,07
B277 1600        LD D,00
B279 5F          LD E,A
B27A D5          PUSH DE
B27B E1          POP HL
B27C 19          ADD HL,DE
B27D 10FD        DJNZ,B27C
B27F D1          POP DE
B280 E5          PUSH HL
B281 C1          POP BC
B282 21C601      LD HL,01C6
B285 19          ADD HL,DE
B286 09          ADD HL,BC
B287 3A775C      LD A,(5C77)
B28A 86          ADD A,(HL)
B28B E1          POP HL
B28C 1891        JR B21F
```

FONT DATA

↓

(NEXT 792 BYTES)

Delete lines 10, 20, and 70. Then rerun the program.

It is very important to understand that you do not change the physical location of RAMTOP by merely POKEing the System variable RAMTOP without entering NEW or by a special routine (see "Automatically Set RAMTOP" page 9 in either TDM, Jul/Aug '85, or "The Best of TDM Vol.1").

Clear the computer by turning it off and back on. If we take the address contained in the System variable E_LINE and subtract the address contained in the System variable VARS we should, according to Figure No.1, find only one byte of memory present in the variables area. Type in the direct command:

```
PRINT (PEEK 16404+256*PEEK 16405)-(PEEK 16400+256*PEEK 16401)
```

Then press ENTER

The result should be a one (1)

To find the value that is stored by the ROM routines, always in the last address of the Variables area, type in the direct command:

```
PRINT PEEK (PEEK 16400+256*PEEK 16401)
```

Then press ENTER

The result should be 128. Turning to page 139 of the ZX81 (page 140 of the TS1000 or page 144 of the TS1500) User Manuals, you will find that the value corresponds to 80 hexadecimal. To prove that this is the marker that tells the SAVE command that it has reached the end of the data to be recorded on tape would require an investigation of the ROM SAVE routine and is beyond the scope of these articles.

This concluded the chapter on the big picture of how your RAM memory is utilized by the BASIC programming system of the computer. Among others, we have dispelled such myths as "you can PRINT on 22 lines of the display", "the programmable area of the screen consists of 23 lines and 32 characters", and "the display file constantly occupies 1K of memory".

TS2050-ZX Modem Modification Project

Dave Clifford

OBJECTIVE: This Hardware/Software modification will allow the user to operate the Westridge TS2050 Modem with the Sinclair ZX Interface 1 and Microdrives.

WHY IT DID NOT WORK: The reason that the TS2050 modem WILL NOT work "as-is" is because of an "address conflict" between the Interface-1 and the TS2050. That is the BASIC reason at least, but it gets slightly more involved when selecting a new Port Address to operate the modem from. For a close-up discussion as to how to avoid this problem, you should consult a copy of "The Spectrum Hardware Manual" by Adrian Dickens (published by Melbourne House). If you have access to a copy, read the section in Chapt. 14 called "Adding 128 I/O Ports". For those of you who are un-able to obtain a copy, I will quote a few words from that section:

"In the Spectrum a very simple form of decoding I/O port addresses is used. A0 is used to select the ULA, A1 the ZX printer and A3, A4, A5 the microdrives and RS-232 interface. No more than one of these address lines should ever be at logic 0 during an I/O operation. If they are then several devices will try to use the data bus simultaneously. Homebrew circuits therefore only have address lines A5, A6 and A7 available to select them".

The stock TS2050 modem uses PORT $73 (HEX) or 115 (DEC) for DATA I/O and PORT $77 (HEX) or 119 (DEC) for CONTROL and STATUS I/O. Convert these numbers to BINARY and you can see where the "rub" is:

ADDRESS LINES

	A7	A6	A5	A4	A3	A2	A1	A0
PORT $73 (HEX) =	0	1	1	1	0	1	1	1
PORT $77 (HEX) =	0	1	1	1	0	0	1	1
					*	*		

* = Conflict on BOTH addresses!

NOW THE "PATCH": Now that we understand what the problem is and we have an idea of what address lines we can not use (A0 thru A4), we now know the "range" for addresses we can employ.

After you install this modification to your TS2050 it will allow you to change the port addresses that the modem uses to those that are a lot more agreeable to the ZX Interface-1 and here is what the new port addresses will be:

DATA PORT = $3F (HEX) or 63 (DEC)
CONTROL/STATUS PORT = $7F (HEX) or 127 (DEC)

And in BINARY...

ADDRESS LINES

	A7	A6	A5	A4	A3	A2	A1	A0
PORT $3F (HEX) =	0	0	1	1	1	1	1	1
PORT $7F (HEX) =	0	1	1	1	1	1	1	1

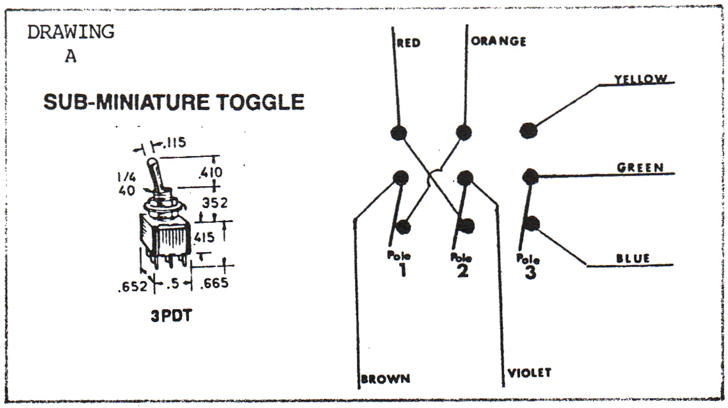
NOW OFF TO THE TOOL CRIB. HERE'S WHAT WE WILL NEED: A Phillips screwdriver (to take the modem case apart), a 25 to 30 watt pencil soldering iron, a pair of needle-nose pliers and diagonal cutters, an X-Acto knife with a nice sharp blade, a drill with a bit that is of suitable size to mount a toggle switch (optional), some solder (quality 60/40 rosin core type), a small wire brush or flux remover, and don't forget some eye protection!

AND THE PARTS DEPARTMENT:

a) 1 three pole double throw switch (Radio Shack #275-6661).

b) 7 pieces of small-gauge "hook-up" wire, approx. 6" long, and you should obtain 7 different colors. The best stuff to use is a length of "rainbow" ribbon cable and separate the conductors into individual wires. The schematic is marked with the first 7 colors that you will normally find. They are Brown, Red, Orange, Yellow, Green, Blue and Violet. (Note: If you decide on a color scheme of your own, be careful! If you make a mistake it could cost you a computer or modem or both.)

READY FOR ASSEMBLY: Now we are ready to open the modem. Carefully remove the case and place the P.C. card on the work bench "solder-side" up. Prepare the toggle switch as shown in Drawing "A". After the switch is complete, study Drawing "B" and "C" (also the schematic) to determine what P.C. conductors are to be cut and where to solder the wires from the switch. When soldering, apply just enough heat and solder to do the job. Now you can select a position on the rear or front panel to mount



your toggle switch. Be sure that it won't get in the way of the P.C. card or a component on the card when you re-assemble the modem. Check and re-check your assembly! If it all looks good, let's go for a test. Can you remember which position the switch is in for "normal mode" and which is the "2050 ZX mode"? First let's try MTERM II (terminal software program) in "normal mode" and see if we are functional (don't connect the Interface-1, it won't work).

If everything is normal, then we are all set to try the "2050 ZX mode". Before we do, we must customize a copy of MTERM II with some simple software POKEs so it will communicate with the TS2050 in this mode.

MTERM II TERMINAL SOFTWARE MODIFICATION: Set up your TS2068 for Spectrum Mode (using an Emulator or Romswitch) and switch it on. Execute CLEAR 53950 and press ENTER. Execute LOAD "mterm"CODE and press ENTER. (Note: If you have a BASIC LOADer that auto-runs MTERM or contains an auto-dialer, remember to change the IN/OUT addresses to the new ones. Also if you are running a print driver that is "port dependant" for data I/O to the TS2050, they too have to be changed. Because of the diverse configurations that are available, you are on your own in this department.) Execute the following POKEs:

I/O Port Addresses and their NEW values.

HEX	DEC VAL	HEX	DEC VAL
E712 =	59154,127	E719 =	59161,63
E724 =	59172,127	E735 =	59189,63
E73C =	59196,127	E73E =	59198,127
E740 =	59200,127	E744 =	59204,127
E74A =	59210,127	E752 =	59218,127
D6A6 =	54950,127	D69A =	54938,127
D67F =	54911,127	D5E1 =	54753,127

The following two POKEs will allow you to EXIT to BASIC while in Spectrum mode and not crash:

POKE 54554,207
POKE 54555,255

Now let's save your "Spectrum Friendly" version of MTERM II to tape. Execute SAVE "mtermzx"CODE 54016,7721 and press ENTER. For the Microdrives, execute SAVE*"m";1;"mtermzx"CODE 54016,7721 and press ENTER. That's it...now you can customize MTERM to your taste. The I/O ports are as follows:

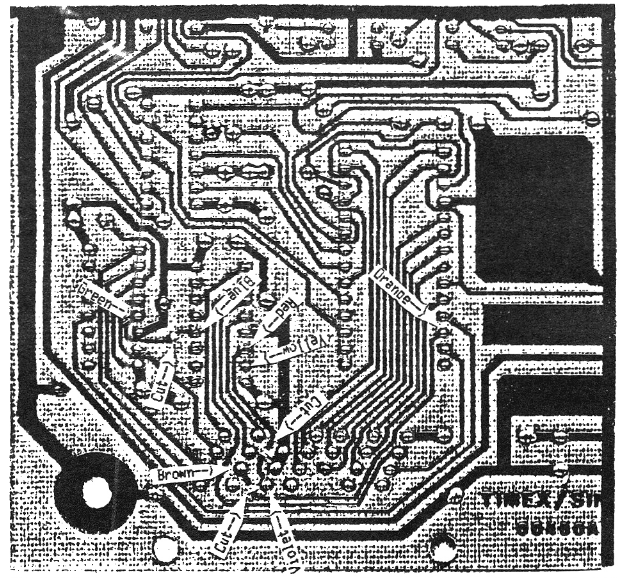
DATA = 63 (DEC) CONTROL/STATUS = 127 (DEC)

PROJECT SYNOPSIS: MTERM II will function just as it did when running on the standard TS2068 except when you "EXIT to BASIC" MTERM's main menu will still be present. To clear the screen just hit the ENTER key and you will have a clear screen with the "K" cursor ready for your commands. Also, with the ZX Interface-1 attached the "extended" BASIC commands and error codes are on-line as well. For example, with Interface-1 attached and you do PRINT USR 54016 to enter MTERM and then EXIT to BASIC, you will see the "Program finished, 0:1" message at the bottom of the screen. And the Buffer Counter will display a slightly different value with Interface-1 attached (30153) as without.

The port addresses that I selected for this project are (to the best of my knowlege) not used by any of the "standard" peripherals associated with the Spectrum computer. If you have a device (like the Aerco parallel I/F) that shares one of the addresses then you may have a problem. You can change the addresses on the TS2050 to just about anything you want. Just remember the binary decoding that the Spectrum and Interface-1 employs and you should have no problems selecting a second address pair.

WHY A SWITCH?? You do not need the switch. The switch allows you to quickly configure the TS2050 to run in Spectrum mode or on say a TS1000/TS1500 with a minimum of components. You can hard-wire your modem for the new addresses and after POKEing the new values into MTERM for Spectrum mode, POKE the same values into a copy of MTERM for the TS2068 (just don't include the two POKEs at 54554 and 54555).

CONCLUSION: Many have enjoyed this modification for months and I hope you will too. If there is enough interest, I have a modification that will allow the TS2050 to dial up the U.K., using the CCITT v.21 protocol.

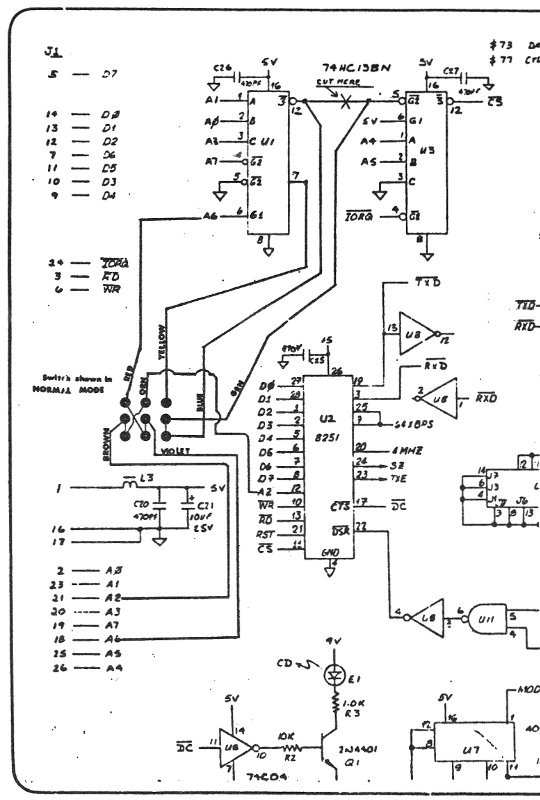


DRAWING B

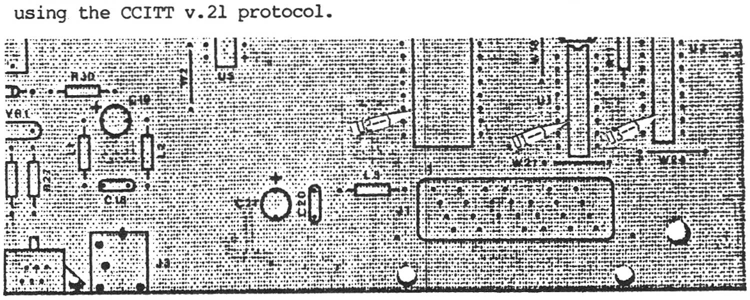
Cut the P.C. "lands" (conductive foil) at the points indicated by the 'Cut-->' flags.

Solder the proper wire from the switch assembly as indicated by the 'COLOR' flags.

CHECK YOUR CONNECTIONS!



DRAWING C



DRAWING D

RLE Graphics (Part Two)

STAN LEMKE

My how time flies when you're having fun!! I dug through the files and was amazed that almost a full year has passed since I submitted the original RLE article (Jan/Feb '87). In that article I explained the use of an RLE DECODER program, a short program that would take an RLE encoded picture, and decode it into a form that our beloved TS2068 could display. Now I would like to explain the other half of the process...RLE ENCODING.

As with the Decoder program, the original Encoder program was written by John Ryan and distributed through CompuServe. I modified the program somewhat, and am presenting it here with John's consent. The RLE Encoder will convert a standard TS2068 screen (256 pixels wide by 176 pixels high) to the RLE format. This is compatible with most of the graphics programs written for the TS2068. The bottom 16 pixel rows are padded with "blanks" to fill out the standard RLE format.

LISTING #1: This is a listing of the RLE ENCODER program entirely in BASIC. It is followed by the CKTYPE output (see May/June '87 TDM). Although the program works, in this form it could well take in excess of an hour to completely encode a complex picture. (Note: The RLE format counts the number of paper dots and ink dots used to construct the picture.) Simply type in the program as listed. Save it to tape with the command RUN 9999 and ENTER. Load and run the CKTYPE program and compare this with the CKTYPE output listing. To RUN the RLE Encoder, type RUN 2020 and ENTER, or LOAD it from tape. You will be prompted to LOAD a SCREEN$ (picture). The encoding will begin immediately. When completed, you will be asked to NAME and SAVE the RLE Encoded bytes.

LISTING #2: This is a listing of the TIMACHINE COMPILER (by Novelsoft) version of the RLE Encoder program. Type in this version if you have this compiler, save it to tape, and compile it. The TIMACHINE COMPILER "list" output is included after the program listing. If you get anything different, double-check your listing. Save the program/bytes as directed by TIMACHINE. The bottom section of Listing #2 has the BASIC LOAD/SAVE portion of the RLE Encoder program. (Reset your computer before going any further.) Type in this portion of the program, LOAD in the compiled bytes, and SAVE the program with the command RUN 9999 and ENTER. When you run this program by typing RUN and ENTER or LOADing the program, you will be asked to LOAD a SCREEN$ (or picture). Encoding will begin immediately and take 30 to 40 seconds...SAVE this to tape as directed.

Now you are able to SEND some of those great TS2068 SCREEN$ to other computers through the common RLE format bridge!

<u>Listing #1:</u>

```
   1 REM ************************

            RLE ENCODER

   Original program by John Ryan
  Presented with author's consent

  modified by ... S D Lemke
     Lemke Software Development
         2144 White Oak
     Wichita,  Ks.    67207

 ********************************

  10 LET X=0
  20 LET Y=175
  30 LET B=0
  40 LET S=40002
  50 LET W=0
  60 LET Z=0
  70 LET C=176
 100 REM ************************

       FIND PAPER (BLACK) FIRST

 ********************************

 110 LET A=POINT (X,Y)
 120 IF A=0 THEN LET B=B+1
 130 IF 0<A THEN GO TO 200
 140 LET X=X+1
 150 IF 255<X THEN LET Y=Y-1
 155 IF 255<X THEN LET C=C-1
 160 IF 255<X THEN LET X=X-256
 165 IF C=0 THEN GO TO 200
 170 IF B=95 THEN GO TO 200
 180 GO TO 110
 200 LET B=B+32
 210 POKE S,B
 220 LET S=S+1
 230 LET B=0
 240 IF C=0 THEN POKE S,32: LET
S=S+1: GO TO 1000
 300 REM ************************

       FIND INK (WHITE) NEXT

 ********************************

 310 LET A=POINT (X,Y)
 320 IF 0<A THEN LET W=W+1
 330 IF A=0 THEN GO TO 400
 340 LET X=X+1
 350 IF 255<X THEN LET Y=Y-1
 355 IF 255<X THEN LET C=C-1
 360 IF 255<X THEN LET X=X-256
 370 IF C=0 THEN GO TO 400
 380 IF W=95 THEN GO TO 400
 390 GO TO 310
 400 LET W=W+32
 410 POKE S,W
 420 LET S=S+1
 430 LET W=0
 440 IF C=0 THEN GO TO 1000
 450 GO TO 110
1000 REM ***********************

 FINISH OUT STANDARD FORMAT RLE

 ********************************

1010 POKE 40000,71
1020 POKE 40001,72
1030 POKE S,127
1040 LET S=S+1
1050 POKE S,32
1060 LET S=S+1
1070 LET Z=Z+1
1080 IF Z=43 THEN GO TO 1100
1090 GO TO 1030
1100 POKE S,43
1110 LET S=S+1
1120 POKE S,7
1130 LET S=S+2
1135 LET S=S-40000
1140 GO TO 2200
2005 REM ***********************

         RLE  ENCODER  BASIC

 ********************************
2010 REM        ENCODE.RLE
2020 CLEAR 39999
2090 REM       [illegible]
2100 CLS : PRINT "Play your TAPE
 to LOAD SCREEN$.": LOAD ""SCREE
N$
```

```
2110 GO TO 10
2190 REM      SAVE
2200 INPUT #0;AT 0,0;"File Name
= "; LINE n$: SAVE n$CODE 40000,
S
2210 STOP
9999 SAVE "RLE.ENCODE" LINE 2020
```

1	414	33990
10	11	794
20	13	1134
30	11	772
40	15	1461
50	11	793
60	11	796
70	13	1114
100	156	9864
110	10	1278
120	24	1905
130	22	2074
140	13	970
150	26	2436
155	26	2348
160	28	2594
165	22	2078
170	23	2178
180	11	831
200	14	1064
210	5	751
220	13	960
230	11	772
240	48	4057
300	157	9814
310	10	1278
320	24	1987
330	22	1969
340	13	970
350	26	2436
355	26	2348
360	28	2594
370	22	1971
380	23	2147
390	11	779
400	14	1106
410	5	772
420	13	960
430	11	793
440	23	1989
450	11	831
1000	156	10345
1010	22	1642
1020	22	1648
1030	13	1044
1040	13	960
1050	12	901
1060	13	960
1070	13	974
1080	24	2115
1090	12	807
1100	12	926
1110	13	960
1120	11	775
1130	13	962
1135	17	1630
1140	12	1105
2005	153	9435
2010	18	1849
2020	13	1328
2090	12	1244
2100	42	5474
2110	10	664
2190	12	1262
2200	64	5348
2210	2	455
9999	25	2450

Listing #2:

```
   1 REM ***********************

     COMPILED RLE ENCODER

  Original Program by John Ryan
 presented with Author's consent

 modified by ... S D Lemke
    Lemke Software Development
         2144 White Oak
    Wichita,  Ks.    67207

********************************

   2 REM !USR 30000
   3 REM ! LPRINT
   5 REM !INT +c,x,y,b,s,w,z,a
   8 REM ! LIST
   9 REM ! OPEN #
  10 LET X=0
  20 LET Y=175
  30 LET B=0
  40 LET S=40002
  50 LET W=0
  60 LET Z=0
  70 LET C=176
 100 REM ***********************

     FIND PAPER (BLACK) FIRST

********************************

 110 LET A=POINT (X,Y)
 120 IF A=0 THEN LET B=B+1
 130 IF 0<A THEN GO TO 200
 140 LET X=X+1
 150 IF 255<X THEN LET Y=Y-1
 155 IF 255<X THEN LET C=C-1
 160 IF 255<X THEN LET X=X-256
 165 IF C=0 THEN GO TO 200
 170 IF B=95 THEN GO TO 200
 180 GO TO 110
 200 LET B=B+32
 210 POKE S,B
 220 LET S=S+1
 230 LET B=0
 240 IF C=0 THEN POKE S,32: LET
S=S+1: GO TO 1000
 300 REM ***********************

     FIND INK (WHITE) NEXT

********************************

 310 LET A=POINT (X,Y)
 320 IF 0<A THEN LET W=W+1
 330 IF A=0 THEN GO TO 400
 340 LET X=X+1
 350 IF 255<X THEN LET Y=Y-1
 355 IF 255<X THEN LET C=C-1
 360 IF 255<X THEN LET X=X-256
 370 IF C=0 THEN GO TO 400
 380 IF W=95 THEN GO TO 400
 390 GO TO 310
 400 LET W=W+32
 410 POKE S,W
 420 LET S=S+1
 430 LET W=0
 440 IF C=0 THEN GO TO 1000
 450 GO TO 110
```

```
1000 REM ***********************

FINISH OUT STANDARD FORMAT RLE

********************************

1010 POKE 40000,71
1020 POKE 40001,72
1030 POKE S,127
1040 LET S=S+1
1050 POKE S,32
1060 LET S=S+1
1070 LET Z=Z+1
1080 IF Z=43 THEN GO TO 1100
1090 GO TO 1030
1100 POKE S,43
1110 LET S=S+1
1120 POKE S,7
1130 LET S=S+2
1135 LET S=S-40000
1140 STOP
1145 REM ! CLOSE #


 LINE 9:  +0
 LINE 9:  30000 #7530

RT20       30584 #7778
RT21       30592 #7780
RT124      30600 #7788
RT129      30605 #778D
c.......... POSINT    30620 #779C
x.......... POSINT    30622 #779E
y.......... POSINT    30624 #77A0
b.......... POSINT    30626 #77A2
s.......... POSINT    30628 #77A4
w.......... POSINT    30630 #77A6
z.......... POSINT    30632 #77A8
a.......... POSINT    30634 #77AA

TIME MACHINE ©1986 Cameron Hayne

M/C: 620 BYTES
+ 16 BYTES FOR M/C VARIABLES
(BASIC WAS 2055 BYTES)

SAVE "m/c"CODE 64732,620
LOAD "m/c"CODE 30000
```

```
1005 REM ***********************

          RLE -- ENCODER

RLE SAVE/LOAD program...
used with the RLE Compiled BASIC

********************************

1010 REM         ENCODE.RLE
1020 CLEAR 39999: LOAD "RLE.ENCO
DE"CODE 30000
1090 REM         LOAD
1100 CLS : PRINT "Play your TAPE
 to LOAD SCREEN$.": LOAD ""SCREE
N$
1110 RANDOMIZE USR 30000
1190 REM         SAVE
1200 INPUT #0;AT 0,0;"File Name
= "; LINE n$: SAVE n$CODE 40000,
(256*PEEK (30629)+PEEK (30628))
1210 STOP
9999 SAVE "RLE.ENCODE" LINE 1020
: SAVE "RLE.ENCODE"CODE 30000,62
0
```

THE FUTURE OF RLE: The draw-back as far as RLE graphics are concerned is that this format is strictly for Black and White pictures. That is fine for B&W computers...and B&W printers, but the TS2068 is a COLOR computer with no less than 2 different color resolutions! Whant can we do with color graphics?

Well, it's no secret, but COMPUSERVE has developed what it calls GIF (Graphics Interface Format). Note: GIF and Graphics Interface Format are trademarks of Compuserve Incorporated (an H&R Block Company. GIF is a standardization of computer graphics (pictures) which have COLOR! I am signed up at CompuServe as a GIF developer for the TS2068, and let me say that this is going to be quite a challenge...so don't hold your breath for a TS2068-GIF program in the very near future! To be meaningful, the resulting picture will almost certainly have to be an EXTENDED COLOR mode picture. (To my knowledge, two programs currently support this TS2068 video mode: my own PIXEL SKETCH/GRAPHICS EDITOR and Dave Franson's EXTENDED PAINT.) The standard color mode would loose too many colors. Remember, we will be trying to decode color pictures created with the AMIGA and the ATARI ST... that have up to 256 different colors (shades) and a wide variety of contrasts. Also, we will be squeezing 640 X 400 pixel graphics into a 256 X 192 picture, or more likely use a window technique to view a portion of this large picture! There is also an added problem of "decompressing" the GIF data. The data may require 32000 bytes of memory in an ATARI ST to display a single picture. This is encoded in GIF to about 6000 bytes, and then further compressed to about 3000 bytes to save upload and download time (and expense). So, first we will have to develop the compression and decompression programs to get at the real GIF data, then develop the decoder and encoder, and finally figure out the best way to display the end result picture (if the TS2068 has enough RAM to handle all this). So, if/when I get this all done, you'll see it right here in TDM!

Time Designs Tests

A Professional Monitor, a unique CADPAK, and some games!

Mike de Sosa

My apologies!

We were supposed to discuss Digital Precision's *DESKTOP PUBLISHER Special Edition* and the QUANTA Library's *PAGE DESIGNER* this month, but the great facilitator failed to come through with the former, and an updated version of the latter did not arrive in time. C'mon great facilitator! We'll report on these next time, if. . . .

PROFESSIONAL MONITOR * * * * *

Hat's off to COMPWARE for Computer 1's new *PROFESSIONAL MONITOR* which has been described as "the most sophisticated debugging tool available for the QL assembly language programmer." I'm, personally, no great shakes as such a programmer, but I've run some demonstration programs and know enough to compare *PM* with several other top-rated monitors in my possession. *PM* is a real stickout by quite a margin!

In addition to the standard features, and a few unrivalled ones, in its previous version, its new features include a symbolic debugging and management capability; an integral Motorola 68000 two-pass assembler; dual-screen debugging; the capability to assemble single and multiple lines of source code using either keyboard entry or entry from existing files; function key control of monitor clones; and fifteen or so new or modified commands. To those unfamiliar with symbolic debugging, it is just a shorthand methodology to take some of the drudgery out of assembly language programming. There is a lot more that one could say here, but I believe this will be sufficient to whet the appetite of any QL assembly language programmer. From COMPWARE, 57 Repton Drive, Haslington, Crewe CW1 1SA, U.K., telephone (0270) 582301; mention my name and Time Designs' for a possible discount. Sorry, no credit cards: use a post office to deposit funds to Girobank account no. 67 361 9508. How much? I'd guess about $65.

PRO-CAD 3 * * * * *

Datanet Systems excellent *Pro-CAD 3*, is a unique two- or three-dimensional, plotter-ready design tool for the QL-user who wants to visualize a design in two or three dimensions rather than building a 3D model. Like so many other programs today, *PC3* was written in SuperBASIC and compiled using one of the several programs which convert BASIC to the much faster and more compact machine code.

PC3 is unique in that it employs a dual screen depicting the plan view of a design and, interchangably, an elevation or side view: this is to accurately portray design data in the third dimension or z-axis, which is difficult to do in most similar software.

PC3 is completely menu-driven and, after a bit of getting used to, easy to learn and use. It is also well-documented and has a comprehensive, interactive HELP facility

The Main Mode screen consists of two side-by-side vertical windows depicting the plan view of a design in the right and an elevation or side view in the left. A thin strip at the top lists various major functions which, if selected, drop a pop-down window from the top of the screen. A data window depicting X, Y, and Z dimensions and angles and other data is at the bottom center of the screen. A crosshair-type cursor may be moved between windows.

With this arrangement, it is amazingly easy to draw three-dimensional figures. Once drawn, you may magnify, rotate, zoom in and out, with a comprehensive array of controls, or select a three-dimensional representation with or without perspective. Circles and other stock figures are drawn where they're supposed to be drawn on U.S. (JSU ROM QLs).

Designs may be output to a standard Epson-compatible printer or a wide range of plotters. *PC3* was written by draftsmen for draftsmen. No price information was included with the software nor in advertising. Datanet Systems advises that they are anxious to do business in the U.S.A. Write to Datanet Systems, P.O. Box 121, Luton, Beds., U.K. for further details.

BUNKERED * * *

QUANTA Library's *BUNKERED* is quite an engrossing golf simulation played over eighteen holes. Using three woods, five irons, and a putter (once on the green), choose your club, select an angle, and adjust the strength of your swing accordingly. You may choose various routes to the pin, depending on your derring-do. Try it, you'll like it. Available with QUANTA membership through a QUANTA librarian. Write Tom Bent, 9016 Flicker Pl., Columbia, MD 21045 or telephone (during 7-10pm EST) (301) 730-7187. Sign up for QUANTA!

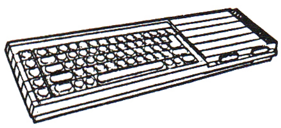
BRIDGE * * * 1/2

Sinclair QL World's Microdrive Exchange offers *BRIDGE* and many other programs at bargain prices. Available at about $10, this contract bridge simulation is quite good with comprehensive options and far, far better than the commercially available *BRIDGEPLAYER II*. But Digital Precision's new *MICROBRIDGE* is on the horizon at about $55. Order *GOLF* using application blanks in Sinclair QL World or send a SASE (envelope) to TIME DESIGNS for a facsimile.

MANDELBROT -- A Fractal World

Michael E. Carver

THIS PROGRAM AND ARTICLE
IS FOR THE SINCLAIR QL



INTRODUCTION

Welcome to the magical world of fractal geometry ("the mathematical study of forms having fractional dimension"). Fractal geometry has been used to build complex mountain ranges and "growing" plants by Lucas Films. It has also been helpful in the study of the baffling world of turbulence. This world was developed by Benoit B. Mandelbrot, a researchest at the IBM Research Center in Yorktown Heights, NY. The "numbers" which lie within the "fractal plain" are said to be within the Mandelbrot set (see solid black area in fig. 1). The set is created and explored by repeatedly calculating a set of complex, real and imaginary numbers. A complex number is any number of the form *a+bi*, where *a* and *b* are real numbers and *i=square root of -1*. A real number is one in which there is no imaginary part. An imaginary number is a complex number with *b* not equal to zero. (A full discussion of imaginary and complex numbers is beyond the scope of this article and knowledge of the author.) By using an iterative process, one can generate the Mandelbrot set. Numbers which do not "fly-off" to infinity are said to fall within the Mandelbrot set. This is where the computer comes in -- the Number Cruncher! For high resolution, a maximum of 1000 iterations is required. In order to fit the resulting data into an unexpanded QL, I have chosen a low to medium resolution with an iteration of 255. If after 255 steps the calculated number does not equal 4, it is assumed to ~~lie~~ within the Mandelbrot set (see fig. 4 for example of the calculation).

The real beauty of fractal geometry is discovered by "zooming" into areas along the edge of the Mandelbrot set. An infinite number of "worlds" can be discovered. By assigning colors to the numbers that "fly" away from the set, one can find violent, yet beautiful, vortices (see figs. 2 & 3). One can also discover minature replicas of the parent set, along with drastic metamorphoses.

THE ADVENTURE BEGINS

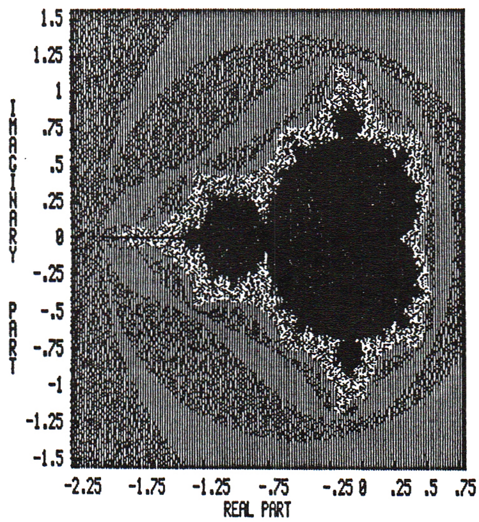
To start your exploratory adventure into the Fractal World, you must first set the parameters of the search. (NOTE: As written, the program expects to be loaded and run from Microdrive #1. Use the command <LRUN mdv1_mandelbrot>.) The selection of "Define New Parameters" from the main menu will allow you to set your sights and the magnification scale needed for zooming into this magical world. First, set the starting corner for the search. A-Corner [real part] is the starting point along the x-axis of fig. 1, B-Corner [imaginary part] is the y-axis point. The size of the search is the length of the sides (a square starting at A & B-corner). When the length of a side is around .01025 or smaller, the resolution of the program begins to fall off. You will need to specify the device (microdrive, floppy disk, etc.) and a name for saving the resulting map. To keep the

fig. 4

```
gap=side/200
for m=0 to 199
  for n=0 to 199
    real=n*gap+a_corner
    imaginary=m*gap+b_corner
    zr=0: zi=0: count=0
    repeat loop
      r=zr^2-zi^2+real
      zi=2*zr*zi+imaginary
      zr=r
      size=zr^2+zi^2
      if size>=4 then exit loop
      count=count+1
      if count>=255 then exit loop
    end repeat loop
  end for n
end for m
```

NOTE: Lines in italics are done via machine code

fig. 1



default device and/or name, simply press RETURN. (NOTE: Remember to include the underscore "_" when naming the device.) After checking to see if the information is correct, you are whisked off on your trek to that particular corner of the fractal world.

While the program is calculating and plotting the selected map, the ESC key will provide you with a report on the status and a mini-menu. The mini-menu will allow the saving of data in progress or a fresh start. (NOTE: Those which cannot operate their QL in F1/Monitor mode may miss the first 8 or so lines of the plotted set -- the ESC key will keep you informed of your whereabouts.) When saving the data and screen, please be sure your device and medium is ready. If at any time you find yourself out of the program, due to an ERROR or by pressing BREAK, you can return by entering as a direct command <main_prog>. DO NOT Run the program, as this will clear the variables used to Call the machine code routines.

Once the computer has completed calculating all 40,000 points in the plane, a prompt will appear asking you to ready the device for saving. A directory of the device is provided, to help avoid overwriting data you may wish to keep. By answering "Yes" to the prompt, you can overwrite the files if you so desire. Another caution on saving: Insure that the medium will have enough space

fig. 5

SUGGESTED "HUNTING GROUNDS"

Real Corner	Imaginary Corner	Side	
-2.25	-1.5	3.0	(see fig. 1)
-0.19920	1.01480	0.05227	(see fig. 2)
-0.95	0.23333	0.06667	
-0.713	0.49216	0.22213	
-1.781	0.0	0.013	(see fig. 3)
-0.75104	0.10511	0.01025	
-0.74758	0.10671	0.00108	
-0.74591	0.11196	0.00143	

for all of the data, approximately 147 sectors total. The iterative mathematics is done via a machine code routine to avoid the days of computer-time which would be required by BASIC. If one were to calculate a fractal plane which lay entirely within the Mandelbrot set, a total of 71,400,000 calculations would be required.

Once a completed picture and data of an area is stored, it can be recalled at any time by selecting "Load Screen and Data" option from the main menu. (Surprise! Surprise!) This option will also re-load a partially calculated plane, allowing one to continue development at a later time.

The color assignments which are provided as a default will not always provide a pleasing result. Therefore, option "Re-Define Color Data" provides a means to select a more satisfactory palette and distribution. Before re-coloring a map for the first time (or one that has just been loaded), the distribution of points must be calculated. It will take about 8 minutes to run through all 40,000 points. (NOTE: Once calculated, this data does not need to be re-calculated before re-defining the colors, as long as the current map is still in memory.) After this array has been amassed, the highest and second highest occurrences of a single point is listed, along with an average. These are only guide-posts to help you in selecting a color spread. Through experimentation, one will be able to use these numbers wisely. You will be asked for a desired range. This is the number of points which will fall within a single color. Next you are asked to provide the desired 7 colors to represent the various degrees of "closeness" to the Mandelbrot set. Quite amazing and different results can be achieved with various mixes of colors and range.

To see the results of a new color selection, choose the "Re-draw Screen" option. The re-plotting of the Mandelbrot map is done via machine code, thus cutting a twenty-minute job in half. After the screen has been re-drawn, a mini-menu will appear allowing you to save the new data, return to the main-menu, or simply sit back and enjoy the results. (NOTE: During the re-plotting of the map, the ESC key and any other keys are disabled.)

PUTTING THE PIECES TOGETHER

There are four separate program listings which need to be entered into the computer before your fractal-adventure can begin. Listings 2 - 4 will compose the different machine code routines used by the BASIC listing 1. Carefully key-in listings 2 - 4 and save before running them. Each of these machine code listings will automatically save the code to microdrive #1, so insure that there is a suitable cartridge in that drive before running them. The BASIC listing should be saved on the same cartridge (or disk) as the supporting machine code routines. (NOTE: Save Listing 1 with the name "Mandelbrot".) The main BASIC listing will automatically load the needed machine code routines when executed. Once you have successfully run and de-bugged all aspects of the program, you may safely delete the BASIC machine code loaders (listings 2 - 4 ONLY).

TOWARDS THE FUTURE

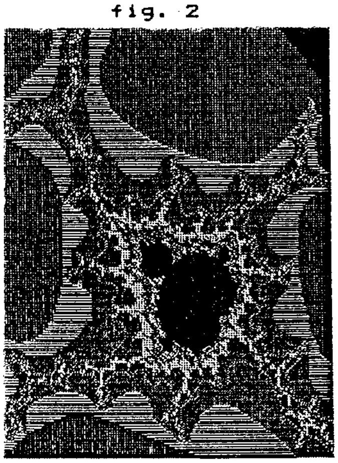
A complete break-down of the BASIC program along with extensive information on the machine code routines (including source code) will be provided in your next issue of Time Designs Magazine.

Further Reading:

1. Mandelbrot, Benoit B. *The Fractal Geomety of Nature.* New York; W.H. Freeman and Company, 1983.
2. Peitgen, H. -O. and P. H. Richter *The Beauty of Fractals; Images of Complex Dynamic Systems.* Berlin, New York; Springer-Verlag, 1986.
3. Dewdney, A. K. "Computer Recreations: A Computer Microscope Zooms in for a Look at the Most Complex Object in Mathematics." *Scientific American.* August 1985. pages 16-24.
4. Schroeder, Peter B. "Plotting the Mandelbrot Set." *Byte.* December 1986. pages 207-210.
5. Nachbaur, Fred "Mandelplot: Mathematical Print Art." *SuncWare News.* May-June 1986. pages 11-14.
6. McDermott, Jeanne "Geometrical Forms Known As Fractals Find Sense in Chaos." *Smithsonian*, December 1983. pages 110-117.

The author will provide the complete program on micro-cartridge for $7.50 (or $4.00 if the sender provides a blank cartridge). Send check or money order to: Michael E. Carver 1016 N. E. Tillamook Portland, OR 97212

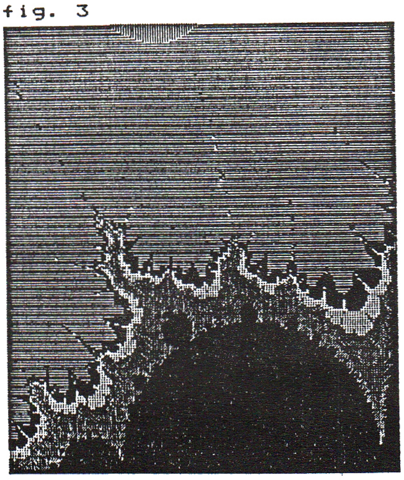
fig. 2



LISTING 1

```
10 REMark **                      **
20 REMark ** Mandelbrot BASIC     **
30 REMark ** Michael E. Carver    **
40 REMark **                      **
50 :
60 initial
70 main_prog
80 :
1000 DEFine PROCedure main_prog
1010   REPeat bigloop
1020     main_menu
1030     store_it
1040     done
1050   END REPeat bigloop
1060 END DEFine main_prog
1070 :
1080 DEFine PROCedure back_door
1090   FOR m=mm TO 199
1100     FOR n=nn TO 199
1110       k=CODE(INKEY$)
1120       IF k=27 THEN mini_menu
1130       cr=n*gap+ac: ci=m*gap+bc
1140       CALL (start+14)
1150       CALL (plotter+704)
1160     END FOR n
1170     nn=0
1180   END FOR m
1182   mini_window
1184   PRINT #8;"To Save --"\"Ready ";dev$\" Press any key"\"To
Continue"
1186   a$=INKEY$(-1)
1187   paste
1190 END DEFine back_door
1200 :
1210 DEFine PROCedure done
1220   LOCal key
1230   REPeat done_loop1
1240     snap_shot: mini_window
1250     PRINT #8;"ESC = View"\"F1 = Main Menu"
1260     REPeat done_loop2
1270       key=keyin1
1280       SELect ON key
1290         ON key=27: EXIT done_loop2
1300         ON key=232: CLOSE #8: EXIT done_loop1
1310       END SELect
1320     END REPeat done_loop2
1330     paste
1340     key=CODE (INKEY$(-1))
1350   END REPeat done_loop1
1360 END DEFine done
1370 :
1380 DEFine PROCedure store_it
1390   REPeat check1
1400     overwrite=0
1410     OPEN_NEW #7,dev$&"dir_copy"
1420     DIR #7,dev$: CLOSE #7
1430     OPEN_IN #7,dev$&"dir_copy"
1440     INPUT #7;na$: INPUT #7,na$
1450     REPeat check2
1460       IF EOF (#7) THEN EXIT check2
1470       INPUT #7;na$
1480       IF LEN(na$)>LEN(name$)
1490         IF na$(1 TO LEN(name$)+1)=name$&"_": check_dev: EXI
T check2
1500       END IF
1510     END REPeat check2
1520     CLOSE #7: DELETE dev$&"dir_copy"
1530     IF overwrite=0: EXIT check1
1540   END REPeat check1
1550   BORDER #8,2,0: CLS #8
1560   CLOSE #8: IF keep THEN paste
1570   mm=m: nn=n
1580   SBYTES dev$&name$&"_scr",131072,32768
1590   SBYTES dev$&name$&"_dat",start,40710
1600   OPEN_NEW #7,dev$&name$&"_dat2"
1610   PRINT #7;PEEK_L(start+642)-start
1620   PRINT #7;mm: PRINT #7;nn
1630   PRINT #7;gap: PRINT #7;ac: PRINT #7;bc
1640   FOR xx=0 TO 255: PRINT #7,color(xx)
1650   CLOSE #7
1660   mini_menu
1670 END DEFine store_it
1680 :
1690 DEFine PROCedure check_dev
1700   mini_window
1710   PRINT #8;"Overwrite?"\"(y or n)"
1720   a$=INKEY$(-1)
1730   IF a$=="y"
1740     DELETE dev$&name$&"_scr"
1750     DELETE dev$&name$&"_dat"
1760     DELETE dev$&name$&"_dat2"
1770     overwrite=0
1780   ELSE
1790     CLOSE #7:DELETE dev$&"dir_copy"
1800     CLS #8: INPUT #8;"New device "\ dev$
1810     CLS #8:  INPUT #8;"New Name "\ name$
1820     overwrite=1
1830   END IF
1835   paste
1840 END DEFine check_dev
1850 :
1860 DEFine PROCedure mini_window
1870   keep=0
1880   IF m>=164 THEN keep=1: snap_shot
1890   OPEN #8,con_192x36a56x0_32
1900   BORDER #8,2,0,5: CLS #8
1910 END DEFine mini_window
```

fig. 3



```
1920 :
1930 DEFine PROCedure main_menu
1940   LOCal  a$,b$,x,y
1950   tv: PAPER 4: INK 7: BORDER 4,2,4: CLS: OVER 0
1960   CSIZE 3,0: STRIP 1: CURSOR 134,18: PRINT "              "
1970   CURSOR 128,14:   STRIP 2: PRINT  "MANDELBROT"
1980   CSIZE 2,0
1990   RESTORE 2080
2000   x=32: y=52
2010   FOR i=1 TO 4
2020     a$=FILL$(" ",2): b$=FILL$(" ",23)
2030     STRIP 1: CURSOR x,y: PRINT a$: CURSOR x+72,y: PRINT b$
2040     READ a$,b$
2050     STRIP 2: CURSOR x-6,y-4: PRINT a$: CURSOR x+66,y-4: PRI
NT b$
2060     y=y+24
2070   END FOR i
2080   DATA "F1"," Define New Parameters ","F2"," Re-Draw Screen
  "
2090   DATA "F3"," Load Screen and Data   ","F4"," Re-Define Colo
r Data   "
2100   REPeat response
2110     key=keyin1
2120     SELect ON key=232,236,240,244 : EXIT response
2130   END REPeat response
2140   SELect ON key
2150     ON key=232: set_up
2160     ON key=240
2170       recall
2180       IF back=232 AND m<199 AND n<199
2182         back_door
2185         RETurn
2190       ELSE main_prog
2200       END IF
2210     ON key=244: re_color
2220     ON key=236: redraw
2230   END SELect
2240   SELect ON key
2250     ON key=232
2260       BORDER 6,0: CLS: canvas: CLS #2
2270       back_door
2280     ON key=100: RETurn
2290     ON key=REMAINDER : main_prog
2300   END SELect
2310 END DEFine main_menu
2320 :
2330 DEFine PROCedure mini_menu
2340   LOCal key,key$
2350   mini_window
2360   PRINT #8,"row ";m;" ";"col ";n
2370   PRINT #8, "F1 = Save"\"F2 = Cont"\"F3 = Restart"
2380   REPeat query2
2390     key=keyin1
2400     SELect ON key
2410       ON key=232: store_it: mini_window: EXIT query2
2420       ON key=236: EXIT query2
2430       ON key=240
2440         CLS #8: PRINT #8;"Okay to"\"Abandon?"\\"  (y or n)"
2450         key$=INKEY$(-1): IF key$=="y" THEN main_menu: EXIT
query2
2460         EXIT query2
2470     END SELect
2480   END REPeat query2
2490   BORDER #8,2,0: CLS #8
2500   CLOSE #8
2510   IF keep THEN paste
2520 END DEFine mini_menu
2530 :
2540 DEFine PROCedure initial
2550   LOCal x,y,i
2560   tv: PAPER 1: INK 0: BORDER 4,2,1: CLS: CSIZE 3,1
2570   x=64: y=68
2580   FOR i=1 TO 16
2590     CURSOR x,y: OVER 1: PRINT "M A N D E L B R O T"
2600     x=x-1: y=y-(i MOD 2)
2610   END FOR i
2620   CURSOR x,y: INK 2: PRINT "M A N D E L B R O T"
2630   canvas
2640   start=RESPR(40710): LBYTES mdv1_mandelbrot_code,start
2650   camera=RESPR(1808): LBYTES mdv1_snapshot_code,camera
2660   plotter=RESPR(786): LBYTES mdv1_plotter_code,plotter
2670   DIM color(255)
2680   color(0)=4
2690   FOR i=1 TO 58 STEP 2
2700     color(i)=6: color(i+1)=4
2710   END FOR i
2720   color(59)=5: color(60)=5
2730   FOR i=61 TO 66: color(i)=4
2740   FOR i=67 TO 82: color(i)=3
2750   FOR i=83 TO 89: color(i)=2
2760   FOR i=90 TO 102: color(i)=1
2770   FOR i=103 TO 118: color(i)=6
2780   FOR i=119 TO 139: color(i)=5
2790   FOR i=140 TO 162: color(i)=4
2800   FOR i=162 TO 192: color(i)=3
2810   FOR i=193 TO 229: color(i)=2
2820   FOR i=230 TO 254: color(i)=1
2830   FOR i=0 TO 255: POKE (plotter+208+i),color(i)
2840   dev$="mdv2_": name$="mandelbrot"
2850 END DEFine initial
2860 :
2870 DEFine PROCedure set_up
2880   LOCal devi$,name1$,key
2890   REPeat query
2900     PAPER 2: BORDER 4,2,5: CSIZE 2,0: CLS
2910     AT 2,0
2920     INPUT "Co-ords of proposed Mandelbrot"\"set"\\"A-Corner
 [Real Part] ";ac
2930     INPUT \"B-Corner [Imaginary Part] ";bc
2940     INPUT \"Length of each side ";s
2950     INPUT \\"Device Name "&dev$\devi$
2960     INPUT \"Save as "&"'"&name$&"'"\name1$
2970     IF devi$<>"" THEN dev$=devi$
2980     IF name1$<>"" THEN name$=name1$
2990     CLS
3000     PRINT\\"Real Part = ";ac
3010     PRINT\"Imaginary Part = ";bc
```

```
3020     PRINT\"Side = ";s
3030     PRINT\"Saving as "&"'"&dev$&name$&"'"
3040     PRINT\\\"Are you satisfied with the above?"
3050     key=keyin1
3060     SELect ON key=89,121: EXIT query
3070     END SELect
3080   END REPeat query
3090   gap=s/200: mm=0: nn=0: calc=0
3100   CALL (start)
3110   array=start+710
3120   POKE_L (plotter+188),array: POKE_W (plotter+192),0: POKE_W (plotter+194),0
3130 END DEFine set_up
3140 :
3150 DEFine PROCedure recall
3160   LOCal dev1$,name1$,key
3170   PAPER 5:INK 0: CLS: AT 2,0
3180   INPUT 'Load from Device ? "'&dev$&'"'\dev1$
3190   IF dev1$<>"" THEN dev$=dev1$
3200   CLS: DIR dev$
3210   PRINT: INPUT 'Name ? "'&name$&'"'\name1$
3220   IF name1$<>"" THEN name$=name1$
3225   canvas
3230   LBYTES dev$&name$&"_scr",131072
3240   LBYTES dev$&name$&"_dat",start
3250   OPEN_IN #7,dev$&name$&"_dat2"
3260   INPUT #7,a: array=a+start
3270   INPUT #7,mm: INPUT #7,nn
3280   INPUT #7,gap: INPUT #7,ac: INPUT #7,bc
3290   FOR xx=0 TO 255: INPUT #7,color(xx): POKE (plotter+208+xx),color(xx)
3300   CLOSE #7
3310   POKE_L start+642,array: POKE_L plotter+188,array
3320   REPeat recall_loop1
3330     m=mm: n=nn: calc=0
3340     POKE_W (plotter+192),m: POKE_W (plotter+194),n
3350     mini_window
3360     PRINT #8;"ESC = View"\"F1 = Continue"\"F2 = Main Menu"
3370     REPeat recall_loop2
3380       key=keyin1
3390       SELect ON key
3400         ON key=27: BORDER #8,2,0: CLS #8: CLOSE #8: EXIT recall_loop2
3410         ON key=232,236: BORDER #8,2,0: CLS #8: CLOSE #8: EXIT recall_loop1
3420       END SELect
3430     END REPeat recall_loop2
3440     IF mm>=164 THEN paste
3450     key$=INKEY$(-1)
3460   END REPeat recall_loop1
3470   back=key
3480   IF mm>=164 THEN paste
3490 END DEFine recall
3500 :
3510 DEFine PROCedure redraw
3520   LOCal array1
3530   array1=start+710
3540   MODE 8: canvas : BORDER 6,0: PAPER 0: CLS #2: CLS #0: CLS
3550   POKE_L (plotter+188),array1
3560   CALL plotter
3570   REPeat redraw_loop1
3580     mini_window
3590     PRINT #8;"ESC = View"\"F1 = Save"\"F2 = Main Menu"
3600     REPeat redraw_loop2
3610       key=keyin1
3620       SELect ON key
3630         ON key=27: BORDER #8,2,0: CLS #8: CLOSE #8: EXIT redraw_loop2
3640         ON key=232: key= 100: EXIT redraw_loop1
3650         ON key=236:  BORDER #8,2,0: CLS #8: CLOSE #8: EXIT redraw_loop1
3660       END SELect
3670     END REPeat redraw_loop2
3680     paste
3690     key$=INKEY$(-1)
3700   END REPeat redraw_loop1
3710   IF key<>27 THEN paste
3720 END DEFine redraw
3730 :
3740 DEFine PROCedure re_color
3750   LOCal m,m1,cut_off,check,i,array1,average,total,xx,col(6)
3760   PAPER 2: CLS
3770   REPeat do_calc
3780     IF calc
3790       PRINT "Do you wish to re-calculate Data?"
3800       key$=INKEY$(-1): IF key$<>"y"AND key$<>"Y" THEN EXIT do_calc
3810     END IF
3820     CLS: PRINT "Please prepare a cup of tea as I","will be awhile calculating","requested data. . ."
3830     DIM dat(255): calc=1
3840     array1=start+710
3850     FOR m=0 TO 39999
3860       dat(PEEK(array+m))=dat(PEEK(array+m))+1
3870     END FOR m
3880     high1=1: high2=0
3890     FOR m=1 TO 255
3900       IF dat(m)>high1 THEN high1=dat(m)
3910       IF dat(m)<high1 AND dat(m)>high2 THEN high2=dat(m)
3920     END FOR m: EXIT do_calc
3930   END REPeat do_calc
3940   CLS: PRINT "Highest Range = ";high1\\"Next Highest = ";high2
3950   total=0
3960   FOR m=0 TO 255: total=total+dat(m)
3970   average=total/256
3980   PRINT \\ "Average Range = ";average
3990   INPUT \\"Input desired Range ";cut_off
4000   CLS: DIM col(6)
4010   color_bar: STRIP 2: OVER 0
4020   INPUT \\"2 colors for Furthest Points "\col(0),col(1)
4030   INPUT\ "Next 4 colors (In order progress-ing toward Mandelbrot Set): "\col(2),col(3),col(4),col(5)
4040   INPUT \ "Color for Mandelbrot Set ";col(6)
4050   check=0: m=254: xx=1:yy=6
4060   REPeat outer_loop
4070     m1=m
4080     REPeat inner_loop
4090       check=check+dat(m)
4100       IF check>=cut_off THEN EXIT inner_loop
4110       m=m-1: IF m=0 THEN EXIT inner_loop
4120     END REPeat inner_loop
4130     IF m1-m<=2
4140       xx=xx+(xx=0)-(xx=1)
4150       FOR i=m TO m1: color(i)=col(xx)
4160     ELSE
4170       yy=yy-1
4180       IF yy=1 THEN yy=5
4190       FOR i=m TO m1: color(i)=col(yy)
4200     END IF
4210     check=0
4220     m=m-1: IF m=-1 THEN EXIT outer_loop
4230   END REPeat outer_loop
4240   color(255)=col(6)
4250   FOR xx=0 TO 255: POKE (plotter+208+xx),color(xx)
4260 END DEFine re_color
4270 :
4280 DEFine PROCedure color_bar
4290   LOCal x
4300   FOR x=0 TO 7
4310     PRINT " ";x;"=";: STRIP x: OVER 0: PRINT " ";:OVER 1
4320   END FOR x
4330 END DEFine color_bar
4340 :
4350 DEFine FuNction keyin1
4360   LOCal a$,a
4370   REPeat key1_loop
4380     REPeat waitloop: a$=INKEY$: IF a$="" THEN EXIT waitloop
4390     REPeat getkey: a$=INKEY$: IF a$<>"" THEN EXIT getkey
4400     a=CODE(a$)
4410     RETurn a
4420   END REPeat key1_loop
4430 END DEFine keyin1
4440 :
4450 DEFine PROCedure snap_shot
4460   CALL camera
4470 END DEFine snap_shot
4480 :
4490 DEFine PROCedure paste
4500   CALL (camera+40)
4510 END DEFine snap_shot
4520 :
4530 DEFine PROCedure tv
4540   MODE 8: WINDOW 512,256,0,0
4550   PAPER 0: CLS
4560   WINDOW 420,160,46,0
4570   WINDOW #2,420,160,46,0
4580   WINDOW #0,420,32,46,160
4590   PAPER 2: PAPER #2,1: PAPER #0,0
4600   INK 7: INK #2,7: INK #0,7
4610   CLS: CLS #0
4620 END DEFine tv
4630 :
4640 DEFine PROCedure canvas
4650   WINDOW #2,400,200,56,0
4660   SCALE #2,199,0,0: PAPER #2,0
4670 END DEFine canvas
```

LISTING 2

```
100 REMark ** Loader for mandelbrot
110 :
120 start=RESPR(712)
130 :
140 REMark *** POKEs Code for mandelbrot
150 :
160 RESTORE 1000
170 FOR x=0 TO 642 STEP 2
180   READ a: POKE_W (start+x),a
190 END FOR x
200 :
210 SBYTES mdv1_mandelbrot_code,start,712
220 :
```


Continued On Page 28.

ANNOUNCING...

800K 2068 DISKS

ONLY $299...COMPLETE!!

A truly COMPLETE 2068 disk system, which includes the following:

Units come to you COMPLETELY assembled & tested!

LARKEN 2068 DISK INTERFACE (with NMI save!)

LKDOS Cartridge with Extended BASIC commands

1 QUAD density disk drive (stores 800K per disk)

Disk case and power supply, and all cables. Disks are 5.25" DS/DD.

Send all orders to:
Please add $10 shipping/insurance
Variety Sales
Includes 90 day warranty.
325 W. Jersey St., #2D
Elizabeth, NJ 07202
☎ 201-527-0535 Catalog $1

In other words, you get a complete ready to run disk system! Just plug in and go!

QL DESKTOP PUBLISHER

FROM: DIGITAL PRECISION

SO MANY FEATURES I DON'T KNOW WHERE TO START! DP'S DESKTOP PUBLISHER HAS 12 INTEGRAL GRAPIC FONTS, 8 INTEGRAL QL FONTS, 150 INTEGRAL GRAPHICS SYMBOLS, 160 INTEG. PAINT BRUSH SHAPES, 10 NON-INTEG. FONTS, UPTO 15X ZOOM, STIPPLES, STRIPES, MIRRORING, ROTATING, IMPORTING OF SCREENS, SELECTIVE SAVES, CUT/COPY/PASTE OVER 2 PAGES, EPSON COMPATIBLE PRINTOUT, FULL FEATURE FONT EDITOR, AUTO-JUSTIFICATION, UNDERLINING, WINDOWING, PAR+SER PRINTERS, CLIP ART, SO NICE TO WORK WITH- "THE CHARACTER CAPABILITIES OF DTP PUT IT IN A LEAGUE OF IT'S OWN", *QL WORLD*. REQUIRES EXPANDED MEMORY & MONITOR TO VIEW. THIS PROGRAM CAN EASILY EMBARASS AN IBM USER WHO PAID 5 TIMES AS MUCH FOR THEIR PROGRAMS! FOR YOU- ONLY $99 !!

2068 specials!!!

Take your pick:

Tomahawk

Machine Code Tutor

Loader V

Critical Mass

ONLY $10 each while current supplies last.

QL Books:

QL SuperBASIC, Definitive Handbook by Jan Jones, creator of SuperBASIC	$20
BASIC Programming on the QL by Neil and Pat Cryer	$18
Good Programming with QL SuperBASIC by Roy Atherton	$15

The above books are in stock for immediate shipment. More books to be available shortly.

Send orders to:

Please add $3 shipping unless specifed other.

VARIETY SALES

☎201 - 527 - 0535

325 W. JERSEY STREET, NO. 2D

Elizabeth, New Jersey 07202

```
220 :
1000 DATA 28672,16890,706,17402,634,8840,20085,28672
1010 DATA 8814,28,-11314,10761,14396,514,8764,611
1020 DATA 29184,9276,611,26880,16878,32,9296,-10802
1030 DATA 16878,24,8784,-11314,13841,-18365,26370,24602
1040 DATA 18938,80,-11031,2,6362,6362,6362,18938
1050 DATA 66,-19820,26408,-19308,26384,16878,32,9296
1060 DATA -10802,20553,-17783,25296,24628,18938,46,16878
1070 DATA 40,9296,-10802,-10775,4,10378,24796,18938
1080 DATA 22,16878,40,9296,-10802,-10775,4,10378
1090 DATA 24776,0,0,0,0,0,0,18938
1100 DATA 488,9850,-14,14555,10459,21579,14555,10459
1110 DATA 30728,16988,23500,-4,29208,12408,282,20112
1120 DATA 8814,88,18938,448,14524,0,18938,460
1130 DATA 15788,0,-26618,11692,2,-26616,15788,0
1140 DATA -26624,11692,2,-26622,32256,12348,14,12408
1150 DATA 284,20112,15798,-26624,-26374,11702,-26622,-26372
1160 DATA 18938,402,15788,0,-26624,11692,2,-26622
1170 DATA 15788,0,-26618,11692,2,-26616,32256,12348
1180 DATA 14,12408,284,20112,15798,-26624,-26374,11702
1190 DATA -26622,-26372,-27908,12,32256,12348,12,12408
1200 DATA 284,20112,18938,322,15772,-26618,11668,-26616
1210 DATA 32256,12348,10,12408,284,20112,18938,322
1220 DATA 14582,-26624,10422,-26622,-27908,12,11593,86
1230 DATA 15804,2050,-26624,11708,16384,0,-26622,18938
1240 DATA 276,15772,-26618,11676,-26616,32256,12348,14
1250 DATA 12408,284,20112,15772,-26618,11668,-26616,32256
1260 DATA 12348,14,12408,284,20112,18938,226,15772
1270 DATA -26618,11668,-26616,32256,12348,10,12408,284
1280 DATA 20112,18938,214,14582,-26624,10486,-26622,19450
1290 DATA 196,15196,0,11100,2,-27908,18,11593
1300 DATA 88,15789,0,-26624,11693,2,-26622,15773
1310 DATA -26618,11677,-26616,32256,12348,14,12408,284
1320 DATA 20112,23625,15789,0,-26624,11693,2,-26622
1330 DATA 15789,0,-26618,11693,2,-26616,32256,12348
1340 DATA 14,12408,284,20112,23881,15798,-26374,-26624
1350 DATA 11702,-26372,-26622,32256,12348,10,12408,284
1360 DATA 20112,32256,12348,4,12408,284,20112,15414
1370 DATA -26624,-27908,22,11593,88,18938,42,14868
1380 DATA 3142,4,27648,14,21061,14469,3077,255
1390 DATA 26112,-422,10362,12,6341,19450,6,10892
1400 DATA 20085,0
```

LISTING 3

```
100 REMark ** Loader for plotter
110 :
120 plotter=RESPR(786)
130 :
140 REMark *** POKEs plotter code
150 :
160 RESTORE 1000
170 FOR x=0 TO 187 STEP 2
180   READ a:  POKE_W (plotter+x),a
190 END FOR x
200 :
210 FOR x=704 TO 784 STEP 2
220   READ a: POKE_W (plotter+x),a
230 END FOR x
240 :
250 SBYTES mdv1_plotter_code,plotter,786
260 :
1000 DATA 18938,190,16980,12820,17914,186,24832,88
1010 DATA 18938,176,16980,17914,164,18426,180,24832
1020 DATA 98,1682,0,1,18938,152,12820,17914
1030 DATA 154,24832,50,17914,140,24832,94,17914
1040 DATA 130,1618,1,3154,199,28618,19450,114
1050 DATA 1621,1,12821,17914,108,24832,10,3157
1060 DATA 199,28588,20085,8814,88,15745,-26624,28680
1070 DATA 32256,12408,284,20112,13558,-26624,9398,-26622
1080 DATA 20085,30208,10322,-10732,-10557,4627,28713,8316
1090 DATA 2,2,30463,20035,20085,17402,532,13146
1100 DATA 0,9050,2,13146,6,9050,8,28720
1110 DATA 30463,8316,2,2,20035,20085
2000 DATA 18938,-514,12820,17914,-516,24832,-614,17914
2010 DATA -532,18426,-516,24832,-598,1682,0,1
2020 DATA 18938,-544,12820,17914,-542,24832,-646,17914
2030 DATA -556,24832,-602,17914,-566,1618,1,3154
2040 DATA 199,28416,12,16978,19450,-586,1621,1
2050 DATA 20085
```

LISTING 4

```
100 REMark *** Loader for Snapshot code
110 :
120 camera=RESPR(80)
130 :
140 REMark *** POKEs Code for Camera
150 :
160 RESTORE 1000
170 FOR x=0 TO 79 STEP 2
180   READ a: POKE_W (camera+x),a
190 END FOR x
200 :
210 SBYTES mdv1_snapshot_code,camera,80
220 :
1000 DATA 8828,1,-4,16890,72,28672,29185,29697
1010 DATA 12505,20994,3074,24,28662,-11524,80,20993
1020 DATA 3073,36,28648,20085,8828,1,-4,16890
1030 DATA 32,28672,29185,29697,13016,20994,3074,24
1040 DATA 28662,-11524,80,20993,3073,36,28648,20085
```

DEALING WITH TRUMP CARD

The 768K RAM Add-on for the QL

by

Mike de Sosa

Miracle Systems' *QL TRUMP CARD* is the last word in disk interface/RAMcard/toolkit hardware for the QL--providing a total of 896K RAM--and another triumph for Tony Tebby. Built-in firmware includes dynamic and static RAMdisk drivers--a shortcoming of the Sandy *SUPERQBOARD* is that the RAMdisk driver is loaded as software, a dynamic printer buffer, a versatile screen dump utility, and a 128K RAM simulator. Facilities are also provided to network QLs enabling them to share printers, disk drives, etc.

WHAT TO DO WITH 896K RAM

With personal computers, RAM is power! The more usable random access memory there is, the more power and flexibility is available for literally dozens of purposes.

Use *TRUMP CARD* to multitask the Psion or Digital Precision suites of software programs, including multiple copies of each if desired, and throw in a desktop publisher or a tedium-reducing game or two. The best ways to do this is to use full-featured multitasking/utility programs like Sector Software's *TASKMASTER*, Tebby's (CARE/QJUMP) *QRAM*, or Compware's new version of *TASK SWOPPER* together with version 3 of Gordon Henson's (COPE) *QL APPLICATIONS TRAFFIC SUPERVISOR (QATS)* and, perhaps, Peter Chambers' (Gap Software) new *FRONT PAGE EXTRA/2*. If you need a good two- or three-dimensional computer aided design/plotter-capable program just add Datanet System's unique *PRO-CAD 3* program or Bob Fingerle's (Tesseract) *CONCEPT 3D*. And there will be plenty of room left for additional utilities such as the useful combination of Peter Batty's (Sector Software) *SPELLBOUND*, Charles Dillon's (PDQL) *FILEBOUND*, and Julian Dyer's (Athene Consultants) new *QL TURBOQUILL+*, with its new glossary and other

features, which really give a leg-up to your favorite (or most hated) word processor. If you're a file freak, load multiple *ARCHIVE*, *RUN-TIME ARCHIVE*, or, better yet if you prefer a ready-designed universal run-time system, Richard Howe's (Ark Distribution) cinch-to-use *ARCHIVIST*, plus miles and miles of files and really relate with those databases. Or, for those extended tea breaks when the spouse is not watching too closely, load up Psion's *QL CHESS*, Mark Steuber's (Sharp's, Inc.) *WAR IN THE EAST*, *SQUADRONS*, QUANTA Library's *BUNKERED*, or Sinclair QL World's Microdrive Exchange version of *BRIDGE* (it's a lot better than *BRIDGEPLAYER II* and a lot cheaper , but don't order the Exchange's version of *GOLF*--it doesn't work properly on U.S. [JSU ROM] QLs.) Whew! I don't usually drop so many names, but it's a quick way to let you know of what programs and program combinations I have found superior.

Back to what you can do with *TRUMP CARD*. Although using disk drives is what the package is all about, you can use *TRUMP CARD's* super fast microdrive imaging (Microdrive to RAMdisk transfer) to load programs and files into QL RAM in a hurry--the quickest way to transfer programs and files from Microdrive cartridges to floppies using *TRUMP CARD* is to first load Microdrive data into RAMdisk using a command such as

FORMAT RAM1_MDV2

and then *COPY*ing or *WCOPY*ing it to a disk. Microdrive programs with corrupt sectors are loaded into RAMdisk and flagged with an '*' in *DIR*ectory listings.

Loading programs, data and document files, and utility files (such as background or foreground programs, datafiles of all types, and such things as key-defined glossary entries) into RAMdisk makes for faster and more pleasurable computing, but don't forget that RAMdisks lose their data in a fraction of an instant when power is cut to the QL, so don't forget to transfer modified files to permanent storage frequently and at the end of sessions--at least until you get a fool-proof battery back-up system for your QL. (Oh, the hours of work I have lost when a Pacific Northwest storm cut power to my QL, even momentarily.) Such battery-backup systems are tricky to design and their price quickly mounts, but I understand that American genius is now being applied to the problem, and we may have a reasonably priced system for use in the States soon.

INSTALLING TRUMP CARD

How do you install *TRUMP CARD* into the QL? Like porcupines make love. Be patient and keep everything parallel and perpendicular. If you have to pull it out and make a reinsertion (this sounds too much like Dr. Ruth) peer inside with a flashlight first to

insure that you haven't bent any of the QL's pins. If you have, it's not the end of the world. Very painstakingly, straighten them out using lots of light and a small-sized but longish screwdriver until they appear to be approximately straight--make several small adjustments rather than taking the chance of going too far. Now reinsert *TRUMP CARD*, as before, and it should slide in like the ignition key to a Porsche. Once inserted all the way, it is worth it. Be patient and you'll most probably get it. (There she goes again.)

USING TRUMP CARD

Individual *TRUMP CARD* features are either on-line or switchable, and, in most cases, transparent (meaning that the QL and the user--you--won't notice they're there.)

Floppy and RAMdisk driver commands, including Microdrive imaging, Microdrive emulation, and a dynamic printer buffer are on-line. A dynamic RAMdisk is created just by accessing it with any of several commands and expands or contracts as data is added or removed from it; a static or fixed RAMdisk

is created by formatting it and it remains the specified size whether data is added or removed. For example,

SAVE ram1_program	If RAM1 does not exist, it is created and the current SuperBASIC program is saved to it as filename; if RAM1 exists, the program is saved to it
FORMAT ram2_80	Creates RAM2 with 80 allocated sectors
*FORMAT ram2_* or *FORMAT ram2_0*	Erase RAM2 with the loss of its data

Convenient Microdrive emulation commands are available on-line, for example

FLP_USE mdv or *RAM_USE mdv*

make the QL think that a program loaded on a floppy or in RAMdisk is on Microdrive--a convenient way of using some programs on different media withou alteration. The commands *FLP_USE flp* or *RAM_USE ram* restore the QLs sense of direction.

The sophisticated printer buffer may be modified for various uses, types of printer, etc. There is no practical limit to the number of buffer files which use the QL's main (expanded) memory.

The screen dump, also rather sophisticated, may be invoked in several ways, including with a defined hotkey of your choice. The entire screen or designated portion of a screen may be dumped to one of 14 types of printers--an inverse image sent to an Epson MX80 is the default setting.

The SuperBASIC command *RES_128* configures your QL to appear unexpanded for the benefit of those programs, e.g., Psion *QL CHESS*, which prefer it that way.

The SuperBASIC command *TK2_EXT* enables *TOOLKIT II* commands and functions which may interfere with some software; if you have a problem with ancillary software, add or remove *TK2_EXT* to/from the software's *BOOT* program and try it again.

QATSWOPPER

Tebby's *QRAM* and Sector Software's *TASKMASTER* have been rather widely discussed and may be the Mercedes-Benz and Rolls Royce of full-featured multitasking/utility programs, but there's a new car on the block, a hybrid made up of Compware's newly upgraded *TASK SWOPPER* which takes the unique approach of providing all the advantages of true multitasking while not gobbling up massive chunks of memory or slowing your QL's pace to a crawl and while remaining compatible with most software (*QRAM* has the rap for being intentionally incompatible with much available software) and Cope's new version 3 of *QATS* (see above), pronounced "cats" because it does away with "mice," a very powerful and sophisticated front-end utility designed to reduce the number of keystrokes required to perform routine (and some not so routine) tasks--shades of old Sinclair single-keystroke BASIC. The two are perfectly compatible--made for each other.

QATS, available on EPROM, Microdrive cartridge, and 3 1/2" or 5 1/4" disk, is fully menu-driven (each menu tailorable to your needs) and is also very economical in both price and memory used, especially if the EPROM version is used. *QATS*, among many other things, also supports columnar and "sideways" printing with its output control.

QATSWOPPER is, thusly, a new lower-cost car on the block. Both programs are straightforward, well-documented, and compatible with *QL TURBOQUILL+*, *SPELLBOUND/FILEBOUND*, *FRONT PAGE EXTRA*, and *ARCHIVIST*.

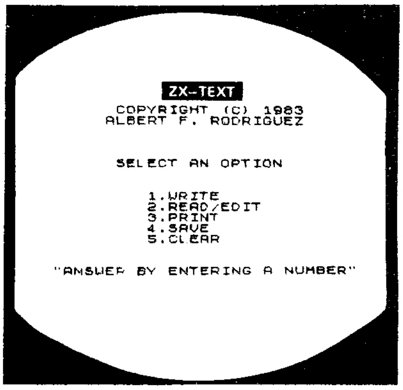
Check with your favorite U.S. distributors first, but, if they don't have the latest versions, don't hesitate to order *TASK SWOPPER*, version 2, and *QATS*, version 3, directly from their distributors, ask for the VAT FREE combined *TASK SWOPPER/QATS* discount price: COMPWARE, 57 Repton Drive, Haslington, Crewe CW1 1SA, U.K. or telephone in the UK (0270) 582301. I'll guess at a combined price of $85, Airmail postage paid, but it could be a bit less.

TRUMP CARD is now available from several U.S. distributors for about $300. Many suggestions for integrating the use of programs such as the ones described above are included in my new book Taking the Quantum Leap: The Last Word on the Sinclair QL, available only from Time Designs.

NEXT TIME: (Depending on software availability) CP/Mulating with the Sinclair QL and more exotic QL software.

POWERFUL AND INEXPENSIVE BUSINESS SOFTWARE FOR ZX81, T/S1000 and T/S1500 COMPUTERS

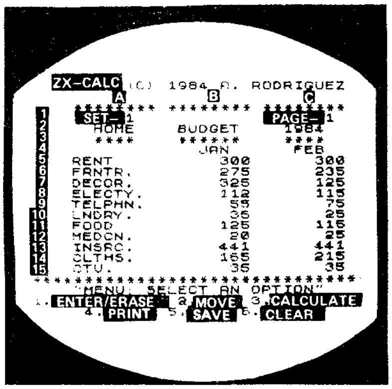
ZX-TEXT



A word processor is to a computer user what a typewriter is to a typist, except that the former has more advantages than the latter. ZX-Text can operate in 16-64K RAM providing from 1300 to 6500 words per document. It features 6 different options: write, read, edit, print, save and clear text. Text is written on a per-line basis with quick speed and with horizontal back-space and delete capabilities being available. You can also access the editor directly from write mode and vice-versa. Text can be proof-read on a per-line basis allowing for enough time to determine if any editing is needed. The text editor allows a line of text to be deleted, inserted, replaced and listed for editing. You may also change a word or expression within a line, stop or start text while it is scrolling up the screen, begin reading text from the first line of the file, re-enter write mode from the editor, return to the main-menu or create a window so that you can read-edit two files simultaneously. The print option takes text displayed in 30-column format on the screen and outputs to either the ZX/TS printer. (With Memotech's Centronics Parallel Interface 80-column and lower/higher - case output is possible.) Files may be saved on tape cassette with the use of one single command, or by the same token they can be erased from memory / storage so that the full capacity of the program can be used for other purposes such as composing letters, reports, articles, memos, standard forms, instructions, ads, graphs, telephone directory, lists of customers, members, friends...etc. Also copies of files are always less expensive and easier to run than using a photocopier. Other advantages are savings in time, paper, ink, correcting mistakes and adding afterthoughts more efficiently than doing them through either handwriting or using a typewriter.

$16.95

ZX-CALC



An electronic spreadsheet calculator is the fundamental basic tool for summarising, reporting and analyzing in matrix form any accounting, mathematical or scientific manipulation of numbers. ZX-Calc operates in 32-64K RAM and affords a maximum of 3360 characters/spreadsheet. The entire matrix consists of 15 columns (letters A-O) and 30 rows (numbers 1-30) with 8 characters/cell. Unlike other popular ESCs, ZX-Calc uses in calculations and within cells all 14 math functions on the ZX-81/TS1000. It offers a unique *SUM function that totals one or more rows/columns simultaneously. Parenthesis can be used within equations. There is no fixed limit on how many equations may be entered. Formulas may be stored in all 420 cells of the spreadsheet. The display affords 15 rows/colums. Loading of data into more than one cell can occur across/down one or more row/column simultaneously. With vertical windowing you can arrange a set of columns in any order, or practice using fixed-variable-alignment display formats. The menu offers 6 options: enter/erase, move, calculate, print, save and clear the spreadsheet. Enter/erase allows the entering, deletion or data alignment within a cell through the use of a mobile cursor. With the move option you may move around the entire sreadsheet to access any row, column or cell. The calculate option allows you to enter labels, values or formulas into a cell or write and enter equations that will act upon the data already within the spreadsheet. You can also enter bar graphs into a cell in this option. Absolute/relative replication, down/across a column/row, is also allowed by this option. Also this option allows the automatic calculation of the entire spreadsheet with one single command. Print allows you to output to either the ZX/TS printer the entire spreadsheet by column-sets and row-pages through use of the COPY command. The entire spreadsheet may be saved on cassette tape or you may clear all data from it or erase the program from RAM entirely. The most salient advantage provided by an ESC over specifically vertical applications software is that an ESC provides a reusable framework with which you can compose any specific financial model rather than just be limited to only one statically fixed format for storing, displaying and manipulating numerical data.

$16.95

ZX-CALENDAR

Time management is an important aspect of any serious business and personal agenda. Planning how to spend our time leaves us better prepared before and while we are spending it and we remain better organized after we finish spending it. ZX-Calendar operates in 16-64K RAM affording 25 appointments in 16K, 100 in 32K or 180 in 48K and 64K. Each appointment record holds a maximum of 220 characters. The main menu includes enter, search/check/sort, change, save, clear and print any and all appointments made on a specific date or with any party. Output to either the ZX/TS printer is permissible. This program will permit you to remember to do something or to be somewhere important by cataloging your answers to six questions that you must account for in order not to waste time when it is scarce: when, with whom, at what time, for how long, where and what are you going to discuss and conclude when you get together with someone else? The program lets you permanently originate, record, classify, search, sort, calculate, modify, summarize, obtain a written report and store your answers to the preceding questions so that you will not forget what you decide to do with your time. This program identifies your time according to when you are going to spend it and with whom you are going to share it. Through these forms of labeling appointments you are able to verify or modify how your time is budgeted without wasting ink, paper or more time trying to remember what you said to yourself or what someone else said to you or where you placed certain written messages that you now can't find. With this program you will know where you can find exactly what you need to know about where you want to and have to be, or where you have been, before you get and after you got there. Thus, ZX-Calendar will let you plan your time so that you will never have to worry about what is ahead or what came before, for you will always know, by using it, to never be caught astray by any time-frame.

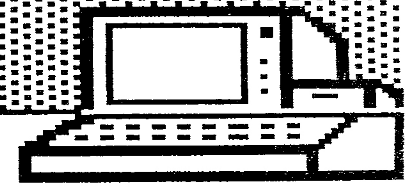
$16.95

$3.00 SHIPPING AND HANDLING/PROGRAM

A.F.R. SOFTWARE - 1605 Pennsylvania Avenue, No. 204 - Miami Beach, Florida 33139

DEALER INQUIRIES WELCOME **(305) 531-6464** FLORIDIANS ADD SALES TAX

TIMEX/SINCLAIR SOFTWARE



WMJ Data Systems
4 Butterfly Drive
Hauppauge, NY 11788
(516)543-5252 11am-6pm EST M-F

**********************SPECIAL OFFER - WHILE SUPPLIES LAST********************

Original Timex Software

ANY TITLE BELOW FOR $3.00
ANY 3 TITLES BELOW FOR $7.50
ANY 6 TITLES BELOW FOR $12.00

Original Timex Software

All titles for the TS1000, TS1500 and Sinclair ZX81 with 16K.

Prod#	Title	Prod#	Title
TX1	ATOR THE ABC GATOR	TX14	GRIMM'S FAIRY TAILS
TX2	AUTO ANALYZER	TX15	INVENTORY CONTROL
TX3	BACKGAMMON AND DICE	TX16	IRA PLANNER
TX4	BUDGETER	TX17	MIXED GAME BAG 1
TX5	CAPITALIZATION MASTER	TX18	PUNCTUATION MASTER
TX6	CHALLENGER	TX19	STATE AND CAPITALS
TX7	CHECKBOOK MANAGER	TX20	STOCK MARKET ANALYZER
TX8	CHESS	TX21	STOCK MARKET GAME
TX9	CONVERSATIONAL GERMAN	TX22	SUPERMAZE
TX10	COUPON MANAGER	TX23	SUPER MATH
TX11	CUBE GAME	TX24	TOTAL TRIANGLES
TX12	FORTRESS OF ZORLAC	TX25	TRAP
TX13	GAMBLER		

Supplies are limited on some titles so please pick alternate titles when ordering. Refunds will be issued for titles out of stock.

Software for the TS2068:

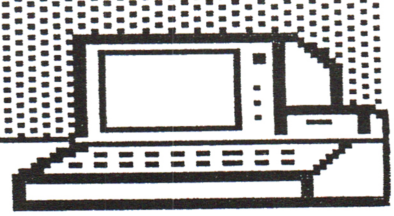
THE CHESSPLAYER (Prod# TS2CP) $13.95. Six levels of play! You can choose either White of Black pieces, change sides in middle of game, printout your moves or board layout to a 2040 printer.

DIAMOND MIKE (Prod# TS2DM) $16.95. This is simply the Best game we have seen for the TS2068. Excellent graphics, speed, sound, music, color... Try this game we guarantee that you will have hours and hours of fun! Joystick optional.

FIGHTER PILOT (Prod# TS2FP) $16.95. Fighter Pilot is a Real-Time flight simulation based on the F15 Eagle, USAAF air superiority jet fighter. This uper simulation offers many of the features found on modern flight simulators including 3D views from the cockpit, fully aerobatic performance, air-to-air combat, crosswinds, turbulance, and blind landing. Joystick optional.

Please include $2.95 per order for shipping. NY residents add sales tax. Please mail your order to the above address. We accept checks, MasterCard, Visa and COD orders. A fee of $1.90 will be added to COD orders. All in stock items will be shipped within 24 hours of receipt of order. Please use Prod# when ordering. Thanks for making us you T/S Software supplier!

TIMEX/SINCLAIR SOFTWARE



WMJ Data Systems
4 Butterfly Drive
Hauppauge, NY 11788
(516)543-5252 11am-6pm EST M-F

ZX PRO/FILE

ZX PRO/FILE (Prod# TS1PF) $19.95 (TS1000). A machine language information storage and retrieval tool for 16-64K. Written by Thomas B. Woods. Multi-word search capability, instant file access, ordered displays, definable printer functions, totally flexible file size, 59 page tutorial manual. This is the finest data base program ever written for the TS1000/1500 and Sinclair ZX81.

ACZ General Ledger

ACZ GENERAL LEDGER 1.000 (Prod# ACZ1) $19.95 (TS1000). An accounting system which is designed to meet the accounting needs of a small business.

ACZ GENERAL LEDGER 2.000 (Prod# ACZ2) $19.95 (TS2068). An accounting system designed for the small business. The ACZ 2.000 is flexibly designed so the reports fit your business whether it is a proprietorship, partnership or corporation. There's plenty of capacity with up to 150 named accounts and up to 800 entries and accounts each month. Includes a 20 page manual that will get you up and going in no time. For use with the TS2068, a 2040 printer is optional.

Colonize the Universe

COLONIZE THE UNIVERSE (Prod# TS2COL) $16.95 (TS2068). Winning this game requires cunning, poise and a bit of luck. Try to build up your space colony in order to survive. Fuel and food are scarce and must be watched constantly in order to survive. Aliens, Black Holes, Super Nova Explosions and Time Warps are also abundant. This game is just like a galactic Monopoly game. It's great fun and provides hours of entertainment! We highly recommend this game!

The Dealer's Den

THE DEALER'S DEN (Prod# TS2DEN) $16.95 (TS2068). If you love to play poker than this is the program for you. You play Draw Poker as you have seen in the arcade games at your local night club. You can also play four hands of Draw Poker against the dealer's one hand. Great Graphics make this software package a must for any computerist who loves Poker! Contains two programs: THE DEALER'S DEN and DRAW POKER.

PRO/FILE 2068

PRO/FILE 2068 (Prod# TS2PF) $29.95 (TS2068). A professional file and database manager for the 2068. Instructions are displayed on the screen and guide you through the steps. After you have saved some files, you can access them by typing in any word, or group of numbers in that file. A 100+ page manual is included detailing the many uses - with instructions for use with a full size printer. The best data base system available for the TS2068. Just ask anyone who uses it.

Please include $2.95 per order for shipping. NY residents add sales tax. Please mail your order to the above address. We accept checks, MasterCard, Visa and COD orders. A fee of $1.90 will be added to COD orders. All in stock items will be shipped within 24 hours of receipt of order. Please use Prod# when ordering. Thanks for making us you T/S Software supplier!

Beginning Z80 Machine Code

PART ELEVEN

Syd Wyncoop

The last instructions we have to discuss are the group that deal with interrupts. This will of necessity be a brief discussion, as it is a most difficult subject and volumes could be written.

In addition, use of the interrupts are very hardware dependent. For this reason, usually the hardware designers and programmers will agree on a few things at the time of system design. Since we already have the system (if I could find the designers, I might hurt them), we are forced to program in the environment provided. This means that we do not have full use of the Z8Ø interrupts. But first, a general discussion of interrupts.

Interrupts are exactly what we would assume them to be, from the name. Whatever the CPU is doing, is interrupted. Now what?

Upon receiving an interrupt, the CPU pushes (saves) all the registers onto the stack and a jump is made to the indicated interrupt service routine (a MC program). We'll find out a little later, how CPU knows where the service routine is.

The first thing most interrupt service routines do is disable the interrupts, to prevent confusion in case another interrupt occurs. Therefore, the service routine must re-enable the interrupt at its conclusion. Since no other interrupts can be accepted until they are re-enabled, the interrupts are commonly re-enabled at the earliest opportunity that won't confuse our friendly CPU.

Interrupts are as competitive as us humans and each one wants to be serviced first. Since the purpose of the interrupts is to provide service for I/O devices, they are normally assigned a priority (by harware design). This allows for orderly service and nested interrupts, much the same as you would have nested FOR/NEXT loops in Basic.

For example, assume our old friend CPU is busily executing your favorite arcade 'shoot-em up' game and you touch the keyboard, here's what happens:

1) An interrupt is generated of priority B, to service the keyboard
2) CPU stops executing your game and jumps to the keyboard service routine
3) During the keyboard servicing, the VDU sends an interrupt of priority C. Since the interrupts have not been re-enabled yet, this interrupt is not acknowledged
4) The interrupts are now re-enabled and the printer sends an interrupt of priority A
5) CPU stops execution of the keyboard service routine and jumps to the printer service routine
6) Assuming no further interruptions, the printer service routine finishes, and CPU returns to the keyboard service routine
7) CPU finishes the keyboard service routine and then returns to your game (you've probably lost a ship by this time!)

Remember, this is illustrative only! Our operating system and hardware are not designed in this manner.

Now, a little more practical. There are two classes of interrupts on the Z8Ø, maskable (MI) and non-maskable (NMI). Maskable interrupts can be disabled (masked) with DI and enabled with EI, while non-maskable interrupts cannot be. Let's discuss the NMI first.

The NMI cannot be masked as DI and EI do not affect it. The reason is that the NMI is a hardware peripheral interrupt that is hard-wired direct to the CPU and is not under software control. Whenever the CPU's NMI line is grounded, a NMI occurs.

The NMI always jumps to 66h for its service routine. This interrupt is normally used for a keyboard or other device that must be serviced immediately. On the TS1ØØØ, the service routine checks for SLOW mode and refreshes the screen accordingly.

However, the NMI is not used on the TS2Ø68. Curiously, a well known bug in the Spectrum ROM was copied over to the TS2Ø68, even though the programmers had obviously tried to make provision for the NMI (see Dr Logan's Complete Spectrum Disassembly). The system variables table tells us that address 5CBØh is not used. If the ROM routine functioned properly, 5CBØh would contain the address of a service routine for the NMI.

Note: Aerco users should be able to fix the TS2Ø68 NMI bug by using the extra ram. I have not had time to try this yet. If I do and it works, I will pass the fix along to Time Designs.

The NMI service routine is properly terminated by a RETN instruction. This allows for properly reseting the CPU (actually the interrupt flip-flop but, we won't discuss that).

The MI is most useful to us. This is a hardware generated interrupt that occurs every 1/60th second in our computers. The TS1000 and TS2068 are initialized in interrupt mode 1 (IM 1) which causes a jump to 38h (RST 38h) for the service routine. The TS1000 uses this service routine to refresh a line of the screen in SLOW mode and the TS2068 uses it to read the key-board. There is a method to intercept this interrupt, on the TS2068, explained in the Timex Technical Manual (available from Time Designs Magazine).

The Z80 has three MI modes, IM 0, IM 1 and IM 2. The primary difference is the manner in which the address of the service routine is determined. As we learned, IM 1 always uses 38h and is the default mode upon initialization of the system.

IM 0 allows any instruction to be placed on the data bus by the interrupting device. This mode is meant to be used with the restarts (i.e. RST 10h), which allow an otherwise three byte Call address instruction to be given in one byte. As we have previously discussed, we cannot use the restarts as they were intended to be used. Therefore, this mode is of limited value to us.

IM 2 uses the interrupt vector (register I) in addition to the byte placed on the data bus by the interrupting device, to determine the address of the service routine. The I register is the MSB and the device provided byte is the LSB. This forms a 16 bit address that points (vectors) to the address containing the address of the service routine, which is usually in a table that provides for the servicing of more than one device.

Use of IM 2 is limited by the lack of pull-up resistors on the data lines of the TS2068. It can only safely be accomplish-ed by using a 'kludge' block of 257 bytes, containing the same data byte. This insures a jump to your vectored address.

Register I is initialized to 1Eh on the TS1000 and is used to determine the character pixel information. The TS2068 initializes I to 3Fh but, I have not determined to what end.

The MI service routine should end with a RETI instruction. This allows for properly returning through nested interrupts and reseting the program counter (PC). You must also re-enable the interrupt with an EI, prior to the RETI, if they have been disabled. The proper sequence of instructions is normally:

```
EI          ;re-enable interrupts
RETI        ;return from interrupt service
```

EI does not re-enable the interrupts until after execution of the instruction following EI. This is to prevent another interrupt from occuring so quick that it confuses the CPU.

Working with the MI is the instruction HALT. Halt does just what you would expect, it suspends CPU's operation until the next interrupt. The CPU will actually be executing NOP's to keep the memory refreshed.

You now have an extremely over-simplified explanation of interrupts and their insructions. The interrupt system is unusable on the TS1000 due to the manner in which the hardware uses it to refresh the screen. The TS2068 provides limited, but useful, access to interrupts.

Since we have learned quite a few new instructions in the above discussion, I have summarized them in the following chart.

Instr	Function
DI	Disables the MI
EI	Enables the MI, after next instruction
RETI	Return from a MI
RETN	Return from a NMI
IM 0	Select interrupt mode 0, forcing service from one of of the restarts
IM 1	Select interrupt mode 1, forcing service from 38h
IM 2	Select interrupt mode 2, forcing service from the vector indicated by interrupt vector and byte on the data bus
HALT	Suspend CPU operations by executing NOP's until next interrupt

I hope this discussion gives some of you hardware hackers the information you need for a nifty project. I'll look forward to seeing your ideas in Time Designs.

You now have all the Z80 instructions. This series has been fun and hope has helped you get started. You can use this series as a MC dictionary but, you need to write programs in order to become conversant in the language.

Next issue I will answer as many questions as I can, as a wrap-up to these lessons. Please send them quickly and direct to me, so that I can have enough time to give a meaningful answer. If I get too many for the next issue, I will answer them individually but, not as quickly.

Syd Wyncoop
2107 SE 155th
Portland, Or 97233

```
;*******************************
;*   Demonstration Routines    *
;*******************************
;
;The following are assorted useful routines that can be
;adapted for use in your programs.
;
;
;Convert Ascii to Binary
;-----------------------
;
;Entry:A=Ascii character
;
;Exit:A=Binary number
;
Asc2Bin Sub "0"             ;remove Ascii bias
        Cp 0Ah              ;is it larger than 9?
        Jr C,EndConv        ;if is, we are done
        Sub 07h             ;else remove more bias
EndConv Ret                 ;A=Binary number
;
;
;
;16 Bit Comparison
;-----------------
;
;Entry:HL=first number
;      DE=second number
;
;Exit:Flags are set as --- zero sign carry condition
;                           1    0    0    HL=DE
;                           0    0    0    HL>DE
;                           0    1    1    HL<DE
;
;This routine always sets zero if HL=DE. Carry indicates which
;is larger, if set then DE>HL, else HL>DE. Sign will indicate
;which is larger, if are signed integers. If sign set, then
;DE>HL, else HL>DE.

Comp16  Or A                ;clear carry
        Sbc HL,DE           ;is HL=DE?
        Ret PO              ;return if so
        Ld A,H              ;invert sign bit if overflowed
        Rra                 ;save carry in bit 7
        Xor 40h             ;complement sign bit 6
        Scf                 ;insure answer <> 0
        Adc A,A             ;restore carry and complimented sign
        Ret
;
;
;String Comparison
;-----------------
;
;Entry:HL=Base of String 1
;      DE=Base of String 2
;
;Exit:Flags are set as --- zero carry condition
;                           1    0    HL=DE
;                           0    0    HL>DE
;                           0    1    HL<DE
;
;This routine always sets zero if HL=DE. Carry indicates which
;is larger, if set then DE>HL, else HL>DE. The strings are
;assumed to be of length <=255 with a preceding byte that
;contains the length.
;
CompStg Ld A,(DE)           ;get length of string 2
        Cp (HL)             ;compare length of string 1
        Jr C,BegComp        ;string 2 is shorter
        Ld A,(HL)           ;string 1 is shorter, get length
BegComp Or A                ;test if length=0
        Jr Z,ChkLen
```

```
        Ld B,A            ;B=# bytes to compare=length of
                          ;shorter string
        Ex DE,HL          ;swap pointers
Loop    Inc DE            ;adjust pointers for next byte
        Inc HL
        Ld A,(DE)         ;get next byte in string 1
        Cp (HL)           ;compare corresponding byte in
                          ;string 2
        Ret NZ            ;return if strings are not equal
        Djnz Loop         ;loop until done
ChkLen  Ld A,(string1)    ;get length of string 1
        Ld HL,string2     ;point to length of string 2
        Cp (HL)           ;set or clear flags
        Ret
;
;
;2068 MC Pause
;--------------
;
;This routine assumes you are using the Rom keyscan and
;will read the system variable that indicates a new key was
;pressed. The only exit from loop is to press a key.
;
Pause   Bit 5,(flags)     ;will be set by Rom keyboard routine
        Ret NZ            ;someone pressed a key!
        Jr Pause          ;loop until key pressed
```

```
;Create IM 2 Kludge Block
;------------------------
;
;This sets up the kludge block referred to in the lesson. It
;fills the area between FE00h and FF00h with the byte FDh.

;You must place a Jp Address instruction at location FDFDh,
;as that is where the IM 2 vector will point.
;
Kludge  Ld HL,FE00h       ;base of kludge block
        Ld BC,00FDh       ;load counter and data
KLoop   Ld (HL),C         ;do fill
        Inc HL            ;adjust pointer
        Djnz KLoop        ;loop to do first 256 bytes
        Ld (HL),C         ;and do last byte
        Ld A,FEh
        Ld I,A            ;put MSB in interrupt vector
        Im 2              ;change interrupt mode
        Ret
```

the CLASSIFIED

FREE ADS FOR SUBSCRIBERS

QL CODEMASTER - QL Cryptography game. 150+ puzzles. Monitor mode only. $12 on mdv, $10 on qd/ds floppy. Add $5 for "C" source code. David Hoshor, 7 Bard Dr., Apt. B4, Hudson, OH 44236.

FOR SALE: QL, mounted in custom fan-cooled case. Centronics interface, Service & Owners manuals, programs: Choice w/ICE ROM, BJ in 3D, Bounder, Sign Designer, Super Astrologer II, Merge 2.0, Screen Dump & 4 Psion programs. Write or call: KRASH, PO Box 180, Schuyler Lake, NY 13457, (315) 858-2918. $300.00.

HELP WANTED: EDITOR(S) for Ver. 3.0 of The Guide to T/S Telecommunications. Pete Fischer has no phone, Steve Ishii now has IBM. Replacements needed. Successful applicants will gain complete control over the project. Contact: Pete Fischer, PO Box 2002, Tempe, AZ 85281.

FOR SALE: 3 SPDOS/SURF N/Ls, 3 T.D., 7 SUMs, 12 SYNCWAREs, ProFile 2068 Manual, SINC.USER 12/86, Sinclair User Annual, Your Computer mag 12/86, 26 BASIC progs. book, BASIC Games book, 8 QuarTerS N/Ls, QL World 2/87. $35 all, $1.50 ea. C. Lytle, 1064 Worth Woods, Worthington, OH 43085, (614) 846-3543.

FOR SALE: TS2068 SOFTWARE. "POKER 4C", a giant 4-reel slot with 4th reel re-spin. Stick or INKEYS. All BASIC and listable. On cassette & includes postage $10.00. Write to: D. Jewett, 63 Charro Dr., Santa Rosa, CA 95401.

WANTED: T/S 1500 COMPUTER and power supply. Write or Call: Mike Maag, 1412 Pleasant View St., Staunton, VA 24401, (703) 885-3577.

WANTED: HUNTER BOARD. Write or Call: Burton Wright, 13622 N.W. Milburn, Portland, OR 97229, (503) 626-8104 (msg.).

CADZ IS HERE! Design on a four screen page. 7 pages with 256K, 1 page with 64K. 33 commands. Fast MC. Requires AERCO disk & ARTWORX. $20.00 from LARRY ZUNK, 4800 E. Cedar Ln., Norman, OK 73017, (405) 366-8595.

JUMP ON THE BANDWAGON. North American Z88 Support Group forming. Send legal SASE for details: Tim Woods, c/o TDM, 29722 Hult Road, Colton, OR 97017.

FOR SALE: Byte Back parallel interface w/driver software $60 ppd. Gorilla Banana parallel printer w/re-inking kit $55 ppd. Ted DuPar, 21696 Cabrosa, Mission Viejo, CA 92691, (714) 583-9763.

NEW SOFTWARE FOR THE 2068/Commodore 1520 printer/plotter: auto screen dump, banner maker, and compatability patch for MSCRIPT Ver.5. Send SASE to John McMichael, 1710 Palmer Dr., Laramie, WY, 82070.

WANTED: TS2068 computer, printer interface, 2050 modem, disc drive interface. Must be very reasonable in price. Ed Salter, 402 Pine Terrace Cir., Milton, FL 32570, (904) 623-8178.

WANTED: Contact from any TS or ZX users in the Fox River Valley (Wisconsin). Is anyone out there? Call Greg at 731-0296 after 5 pm, or write to me at: 2300 N. Clark, Appleton, WI 54911.

FOR SALE: TS2068, TS2020, TS2040, power supplys/AC adaptor, Aerco printer interface, 2 joysticks, 3 books, 20 software (cass.& cart), W.P., bus., game, educ., misc. programs. All exc. cond. with documentation. Will accept any reasonable offer. I will pay shipping costs if buy all. Call or write: Carolyn Bower, 560 Catalina, Wooster, OH, 44691, (216) 345-5337.

* * * C L O S E - O U T S A L E * * *

TS2068 SOFTWARE

"DIAMOND MIKE"
"GREAT GAMES AND GRAPHICS SHOW"
Each originally sold for $24.95, but get both for just $9.95--while supplies last.

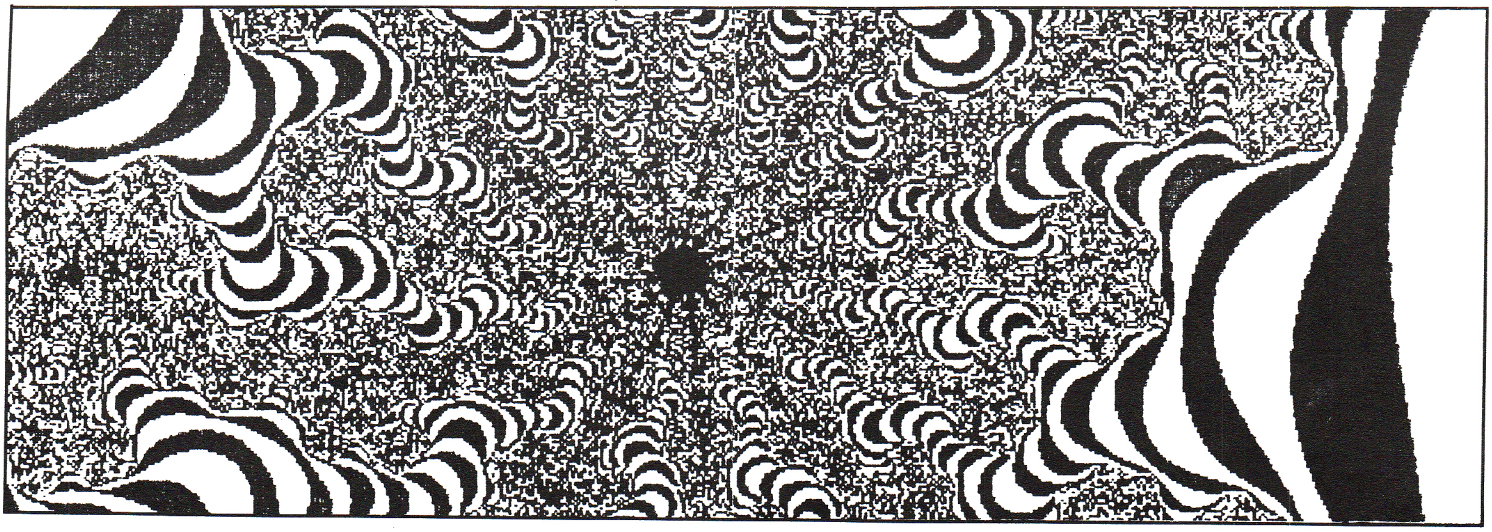
"INTEGER BASIC COMPILER"
"Z80 ASSEMBLER"
Both for $9.95 (while supplies last).
Includes 19 pages of documentation

JRC SOFTWARE
200 N. Main St.
Scottsburg, IN 47170
(812) 752-6071

Do you have some equipment or a program that you would like to sell? Looking for something hard to find? Place an ad in **THE CLASSIFIEDS!** Subscribers can place one free personal ad in each issue. Ad size is 32 Col. wide (like 2040 paper) and maximum of six lines. For additional lines - $3 each. **NON-SUBSCRIBERS** and **DEALERS:** $4 a line. **DEADLINE FOR ALL CLASSIFIED ADS:** Two weeks before publication date. Mail your ad to: **TIME DESIGNS MAGAZINE, The Classifieds Dept., 29722 Hult Rd., Colton, Oregon 97017.**

PROGRAMMING SRAM HI*RES
PART III

By Fred Nachbaur



This is the final episode of a tutorial "mini-series" on the use of SRAM HI*RES EXTENDED BASIC (SHR-EB). This time around, we'll wrap up our coverage of UDG's (User-Defined Graphics), and examine some of SHR-EB's truly unique aspects: Sprites, window-scrolls ("pseudo-sprites"), and the F-SAVE tape routines.

WHAT A DEAL!

First, though, we'd like to make you an "offer you can't refuse." Mr. Gregory C. Harder, main author of the SHR-EB, has developed a machine-code RLE (run-length encoding) decoder for use in conjunction with SHR-EB. What's more, he has written it up in a wonderful tutorial article and has given it into public domain. All he asks is that you supply a 6x9" or 9x12" envelope and $2 for photocopying and postage. I'll honor the same deal through Silicon Mountain. Send to Greg's address as given in the SHR-EB documentation, or to Silicon Mountain Computers.

So what is "RLE" anyway? Essentially, it is a "standard" of data transmission of graphics and pictures. Data encoded in this way is thus transportable to any machine capable of at least 256x192 pixel resolution (as your T/S using SHR-EB), regardless of what machine was used to create the picture. Compuserve and several BBS services have RLE pictures on file, that you can download and decode using Greg's routines. Would you believe the "Japanese Girl" picture originally used to promote MacPaint, on your humble TS1000? Believe it! This an many others (some of them very striking) can now be displayed right on your screen, F-SAVEd to tape, etc.

MORE UDG'S, PLEASE

In the BLACKJACK program, we create the pictures of the cards entirely using UDG's. If you tackle such a project yourself, you'll soon realize that this approach can easily "eat up" more than the 128 UDG characters available in the "MODE 2" UDG buffer. Don't despair! There are ways of getting almost unlimited numbers of UDG's.

The most obvious approach is to define new UDG's, as required, "on the fly." Redefining a UDG or set of UDG's does NOT change what is already on the screen; after printing something using one set of UDG's, simply redefine some or all of your UDG's (perhaps by placing the definitions into a subroutine). In principle, you could fill the whole screen with different UDG characters (768 unique characters) using six different sets of definitions (6*128=768).

However, the amount of memory that would be used by all these definitions, even using the compact hex format, make this approach somewhat impractical. In the case of the BLACKJACK program, I found that I had the following UDG requirements:

3 - 5*8 face cards=3*40 = 120
"Vegas Bob" = 8
Fill character (back of cards) = 1
4 suit symbols = 4
4 upside-down suit symbols = 4
13 card symbols (A,2,3 etc.) = 13
13 upside-down card symbols =13

TOTAL UDG's REQUIRED = 163

Note that just the face cards and "Vegas Bob" use up 128 UDG's, the total number available. Where do we get the remaining 35 required UDG's?

The first UDG-saving "trick" was to scan over the definitions of the face-cards, to find any duplicated definitions. Some of the UDG's used more than once are the ones corresponding to P, Y, and inverse 6. This reduced the number of UDG's needed by 9, allowing me to include the suit symbols (normal and inverted) and the back-of-card fill character. Now I only had to accomodate the 26 card numbers (13 normal, 13 inverted).

For the normal symbols I could have, of course, used the mode 0 or mode 1 print characters. However, I found that the aesthetics were improved by using characters only 5 pixels wide instead of 7 (these were actually "lifted" from the 40-column character set in ZX-TERM*80). So where did I find 26 more UDG's?

The solution was to define 26 of the available 32 SPRITE definitions. This points out the first characteristic of a sprite; it is basically just a special UDG, and can be used analogously to a normal UDG. The differences are as follows:

1: Printing a sprite does not overwrite what is already there; instead, it "ORs" the image with what is on the screen at that location. TS2068 users will recognize this property as the OVER command. So, in a sense, we've added the OVER feature to the TS1000 with the inclusion of sprites in SHR-EB. If you're using sprites merely as a UDG, you have to be sure that nothing is printed at the proposed sprite location. If in doubt, print a space first, then print the sprite at the same location. In the BLACKJACK program, this is not a problem, and is therefore un-necessary.

2: The screen is mapped differently for SPRITE PRINT than it is for normal or UDG PRINT. The reason will become clear when we explore the sprite commands in greater depth. For now, simply remember that the vertical axis is inverted, and the horizontal axis is multiplied by 4. In other words, sprites use PLOT co-ordinates instead of PRINT co-ordinates. This will require a bit of juggling if you're using a sprite as a UDG. A careful study of the PRINT CARD subroutine at lines 10-130 may give some insight into this.

The third UDG-saving trick occurred to me only after writing the BLACKJACK program. Had I thought of it earlier, the coding of the CARD PRINT routine might have been somewhat simpler. However, "I have written what I have written," so I'll simply describe the method for your use in your own programs.

There is no reason why the "mode 1" PRINT mode HAS to be lower-case characters. BLACKJACK does not use this PRINT mode at all, to save you the trouble of first loading a lower-case character set, like the one contained in HR*DEMO*1.

PRINT mode 1 redefines CHR$ 38-63 (A-Z) and CHR$ 128-165 (inv. space-inv. 9). Normally you'd use this for lower-case and custom characters, but there is no reason why you couldn't use these to give you 64 additional UDGs. Printing the inverse alpha characters from mode 1 will still give (normal video) capitals, and the lower graphics and symbols will remain the same, but the rest will now give your additional UDGs.

Note that, in our documentation, we say that "There are no commands from BASIC to change these patterns." In other words, there is no analog to LPRINT U; or LPRINT SD; for this mode. However, you CAN use BASIC (good old PEEK and POKE) to transport UDG definitions into the mode 1 buffer. Proceed as follows:

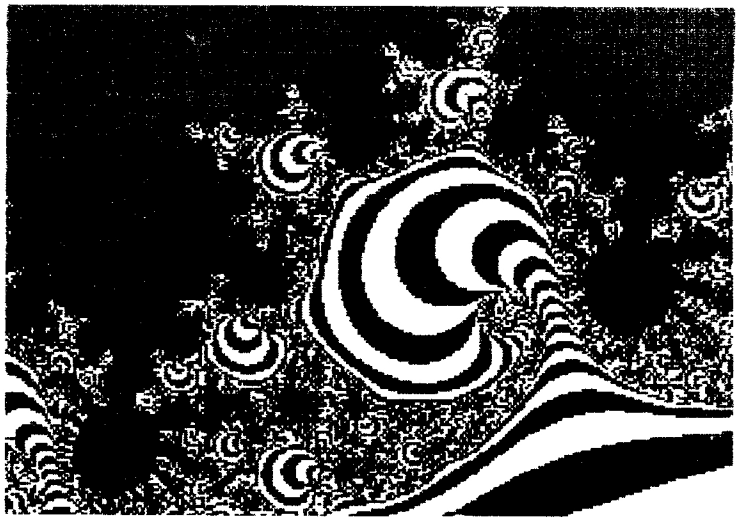
1: Define up to 64 UDG's as usual. You will have to define them in two blocks: the first block runs from A to Z (CHR$ 38-63), and the second block from inverse space to inverse 9 (CHR$ 128-165). These will end up being the "alternate" (mode 1) set. Execute the definitions to store them in the UDG buffer.

2: Move the newly created characters from the UDG buffer (BUF1) into the mode 1 buffer (BUF2). The following routine will do the trick:

```
1000 FOR N=15360 TO 15663
1010 POKE N-512,PEEK N
1020 NEXT N
1030 FOR N=15664 TO 15871
1040 POKE N-1024,PEEK N
1050 NEXT N
(FAST mode recommended)
```

3: Define your "main" UDG set (up to 128 UDG's) as usual.

Your alternate set will now be housed in the mode 1 buffer. Access these using PRINT;;; and access the main set using PRINT;;;; as usual. You now have 192 "true" UDG's at your disposal, without having to resort to sprites. Note that when you re-LOAD your program later, you will have to re-execute the definition lines and PEEK/POKE loop before running the program, to insure that the alternate set as well as the main set of UDGs are stored in their respective buffers in your SCRAM or similar static RAM board. See also the section on F-SAVE MAGIC for time-saving short-cuts.



PHILOSOPHY CORNER

As a lead-in to our discussion of sprites, here's a bit more "SRAM HI*RES philosophy." In designing SHR-EB, we tried to make a package that is useful to the broadest range of users, maximizes flexibility, is immune to crashing because of user error, while maintaining at least some degree of standardization with BASICs for other systems. Quite a tall order; the result, we feel, satisfies all these criteria. However, some features may not be needed by all users, and some aspects may require some degree of compromise.

For instance, with the inclusion of a 64-column PRINT mode, we decided to double the range of the PRINT AT and TAB commands. To reduce confusion, the same horizontal range (0-63) is used for ALL printing modes. In other words, the horizontal co-ordinates are multiplied by two, as compared to the original Sinclair system. Similarly, since we can now start printing on any vertical line, the vertical PRINT AT co-ordinate is multiplied by eight. Still, we followed the Sinclair convention of placing the "origin" (PRINT AT 0,0) at the top left of the screen.

It is a Sinclair quirk that PRINT and PLOT use inverted vertical axes. In other words, the origin for PLOT is at the lower left corner of the screen. Furthermore, the axes are expanded since we can PLOT at twice the resolution of PRINT. When coming up with a high-res PLOT, we decided to retain the Sinclair protocol. After all, we wanted to write a true extension to Sinclair BASIC, rather than "dream up" our own version of how BASIC should operate. As a result, our PLOT co-ordinate system closely parallels that of the Spectrum/TS2068. The only real difference is that we have two screen-size modes (22 and 24 lines). In 22-line mode (IF USR HR THEN CLEAR), the co-ordinates are just like the Spectrum/2068. In 24-line mode (IF USR HR THEN RUN), the origin is moved downward by 16 pixels, giving us a range of 0-191 instead of 0-175.

So we basically have two discrete systems for defining screen locations. The first goes from 0-63 horizontally, and 0-191 vertically starting at the top (PRINT and related commands like LPRINT U;). The other goes from 0-255 horizontally, and 0-191 vertically starting at the bottom (PLOT, DRAW, WINDOW and similar commands). Which system is used depends on the range of the particular command. The only real exception is the REVERSE command (IF USR HR THEN RAND). Since we can only control 32 columns of screen locations for video reversal, and cannot control the effect vertically, the range for this command is only 0-31 horizontally.

THE AMAZING SPRITE

Sprites are mapped like PLOT, starting at the lower left. The reason is that sprites (unlike UDGs) can be placed at ANY horizontal pixel location. Another reason is that SPRITE MOVE shares the same line-drawing algorithm used by DRAW.

The sprite commands are: SD; (sprite define), SP; (sprite PRINT), SE; (sprite erase), and SM; (sprite move). These are all Group 2 commands, so must of course be preceded by IF USR HR THEN LPRINT. Like the other Group 2 commands, they can be concatenated into multiple-statement lines. Without getting into depth on the syntax requirements (read the manual!), here is how you use these commands.

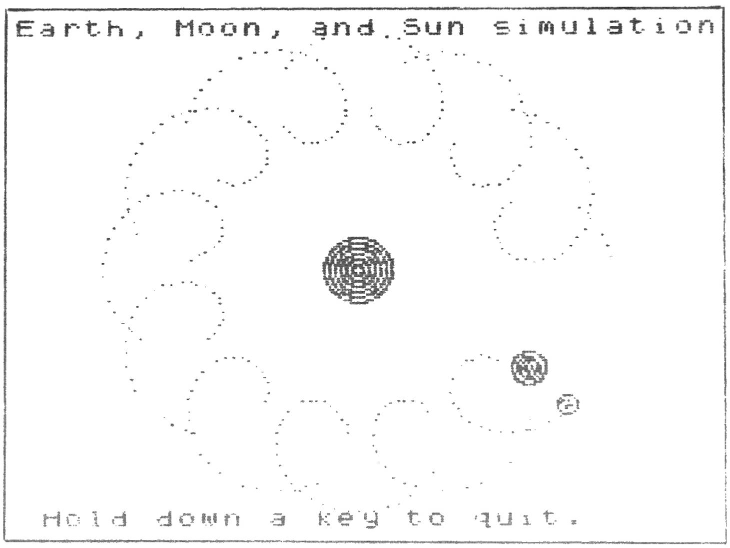
Sprite Define (SD;) is used almost precisely like UDG Define (U;). The only difference is that with UDGs you define the starting character as a string literal (in quotes), whereas sprites do not relate ("correspond") to character sets, and are therefore defined using a numerical sprite number (0-31). The hex sequence that follows the starting sprite number defines the pattern(s) for the sprites (up to 32). As with UDGs, you can get the apparency of a lot more than just 32 sprites by judiciously redefining sprites "on the fly."

Sprite Print (SP;) puts a specified sprite at the specified location on the screen. You are in effect "plotting" the sprite, since as pointed out before, the SP; command uses the PLOT co-ordinate system. The location specified relates to the upper left corner of the sprite pattern.

When using sprites simply as "extended UDGs" as in the BLACKJACK program, these are the only two commands you'll be using. The other two are used when employing sprites for their intended purpose - movable "transparancies." Until now, things you put on the screen were immutable and permanent. In a sense, sprites are "above it all." As mentioned, when printing a sprite, it does not erase what is already there. Instead, it is as if the white portions of the sprite are transparent, allowing what is "below" to show through. Similarly, when removing or moving a sprite, only the sprite pattern is affected; what was there before is left alone.

This is where the Sprite Erase (SE;) command comes into play. This removes the last sprite printed or moved. If you don't invoke SE; before printing or moving the next sprite, the previous sprite becomes a permanent part of the "background plane" or main screen display. You can therefore selectively allow sprites to "melt" into the background, or vanish without a trace.

Sprite Move (SM;) moves a specified sprite around the screen in a specified straight line. Anyone who has viewed the HR*DEMO*2 program supplied with the SHR-EB package needs no convincing as to how impressive this can be. As with Sprite Print you must follow Sprite Move with Sprite Erase to prevent it from "solidifying" into your background.



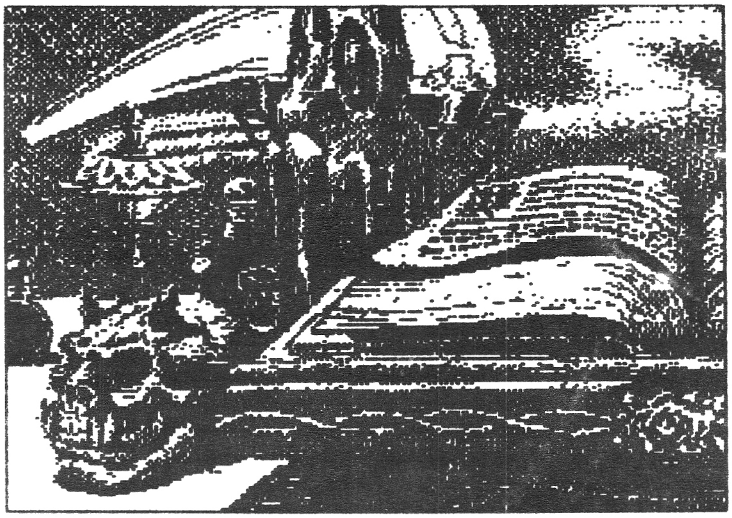
PSEUDO-SPRITES

The biggest limitation with the SHR-EB sprite commands is that they only operate on single 8x8 pixel patterns. What if you want to move larger figures around, a pixel at a time?

The bad news is that true "big sprites" are not possible using SHR-EB. However, you can obtain "pseudo-sprites" using the window-scroll commands (WU; WD; WL; and WR;). Study the "Earth-Moon-Sun" part of the demo to see an example of this. The moon is a true sprite; notice that it does not erase its orbital path as it crosses it. The Earth, on the other hand, is a pseudo-sprite. Like a true sprite, it moves around a pixel at a time. However, since it's in the background plane it erases things that it crosses, as the moon's orbital path.

The one advantage of the pseudo-sprite over the true sprite is that it requires no additional definitions. No matter how you create the pattern to be moved, it will move as a unit when using the window commands. As a point in fact, the "earth" figure in the demo was created using a combination of CIRCLE (multiple use of PLOT) and SPRITE PRINT (the moon was printed twice, one pixel apart, at the center of the earth, to give it that textured look).

You may have to experiment with the exact size of the window to be moved, depending on how far you are moving it, to prevent stray pixels from being left after the move.



F-SAVE MAGIC

What does F-SAVE stand for? I'll leave it up to your imagination. It's a Fast tape routine written by Fred, and I have since given the central algorithms away for Free to others for inclusion in their Fine software. If you're having trouble with F-SAVE, you might come up with other meanings.

However, if you ARE having trouble, don't despair. If it seems to ALMOST work, it usually will only require an inverted load cable to work on your particular system. See the second paragraph on page 18 of the documentation.

Assuming that you've gotten it to work properly, though, I'll discuss some nifty features of the F-SAVE package that you may not have thought about. The first of these is the usefulness of the BOOT program included in the package. This is a shortened version of the F-SAVE "Load" command only. Save a copy of this as the first thing on each of your SHR-EB F-SAVE tapes. On power-up, you can thus load BOOT (takes only about a minute), then immedatiately fast-load your applications programs.

Continued Next Page.

When you load BOOT, it stores a duplicate of the F-SAVE load routines at the usual run-time location, then proceeds to actually run it. As your application program loads, it of course will "overwrite" the copy installed by BOOT. However, since it's an exact copy, the load routine just keeps running!

Once the first SHR-EB application has been loaded, the F-SAVE routines will of course be present in memory, and you can use it to load other programs, variables, or screen data.

The second point relates specifically to the BLACKJACK program. Refer to lines 9100-9150. After loading the program conventionally, you can GOTO 9100 to F-SAVE the program to tape. The first thing that happens is that we POKE location 32600 with a "1". (Why? We'll see later.) The BLACKJACK program is then saved by the command IF USR HR THEN SAVE "BLACKJACK",P in line 9120.

Next, since location 32600 contains a "1", the program saves the entire contents of the 8-16K region, including the current screen, all UDG and sprite buffers, and system variables. The line that follows (9140) is ignored, since location 32600 contains "1".

Since we're saving (and later reloading) the entire 8-16K region of memory, we have no need to redefine all those sprites and UDGs! This means that you can BREAK operation at the cover screen, return to the normal screen display by entering FAST mode, and then delete all lines from 9000-9021. The memory you save thereby can be used to install your own additions and refinements. Then GOTO 9100 to F-SAVE.

Let's analyze what happens when you later reload the program. This will normally be right after power-up or NEW, so location 32600 will contain a "0". This location is right near the top of your 16K memory, specifically 168 bytes below the default RAMTOP. This is just low enough to be out of the way of the stacks (under normal conditions) while being well above the end of the BLACKJACK program. After program loading has completed, the program will auto-run starting at line 9130. Since location 32600 is beyond the end of the program, variables and workspace area (higher than STK_END), it will still contain "0". Line 9130, which would otherwise try to F-SAVE the present (presumably trashed or otherwise not apropos) screen area, is therefore ignored. Instead, it fall through to line 9140, which WILL execute because 32600 does not contain a "1" This causes the DATA (screen and UDG/sprite buffers, etc.) to be loaded from tape. As mentioned before, the UDG and sprite definitions will already be in place, and there is no need to re-define them. The program therefore jumps directly to the cover-screen routine at line 2000.

Once again, SHR-EB manages to impart features to the TS1000 that are suspiciously reminiscent of the Spectrum/TS2068. While not quite as comprehensive as the Spectrum tape routines, F-SAVE does drastically improve the flexibility of the machine by offering various "import/export" options for screen data and/or BASIC variables.

Getting back to the topic of RLE graphics for a moment, here is one possible scenario for manipulating your screen images. After downloading an RLE picture from a BBS, using either ZX-TERM*80 or Mini-Xmodem, follow Greg Harder's instructions for decoding and displaying the picture. The decoded picture can then be F-SAVEd to tape for future viewing. You could then load the "Thrust" variant of Sinc-Artist, and use its swap feature to transfer the picture into Sinc-Artist's A$. You can now use Sinc-Artist to polish up the picture, add or delete features, etc. When done with your graphic manipulations, transfer it back into low memory. You have a full graphics system that compares favourably with MUCH more expensive systems which will remain MacNameless.

A FEW IDEAS

What can you do with SHR-EB? Well, you could start by designing your own "Classy Front Ends" as Paul Bingham did for the 2068. You could even use Paul's character patterns as published in TDM 3:5. Since Paul published the patterns in decimal, you'll have to convert to hex to use them in the SHR-EB "LPRINT U;" command. This is a great opportunity to use the easy dec-hex conversion technique given in Appendix II of the SHR-EB documentation.

Foreign character sets? No problem. Russian, Hebrew, Greek, and Katakana characters are now possible. By studying how the 64-column character set is arranged in the data table at 4E94-5093h you could even come up with foreign 64-column characters. (HINT: each nybble contains a character. Split each group of eight bytes into left and right nybbles.)

How about high-res "big characters?" You could use UDGs to design, let's say, 2x3 Olde English characters. Use the techniques outlined in this article to economize UDGs, and you should be able to design custom characters for the whole alphabet. Use the string array approach as in BLACKJACK to store the UDG codes for each character, and print them out in a loop.

Need a transparent flashing cursor? A sprite would be a natural for such an application. What other applications can you dream up for the Sprite Print, Sprite Erase, and Sprite Move commands?

Machine-coders: Use short LDIR routines to shuffle around various sprite and UDG definition tables. This will give you fast graphic possibilities limited only by your imagination and available memory. Incidentally, the HR*CORE program contains Hot Z names for all the major routines and variables; many of the routines can be usefully called from within your machine-code application.

The possibilities for adaptation of Spectrum and 2068 software are endless. Over the years a great many programs have appeared, that would have looked silly in the old low-res system, but are now quite feasible. Many books of interesting routines exist for other computers, as IBM PC and Apple. These, too, are now "fair game" for your not-so-lowly-anymore ZX/TS.

Thus ends the SHR-EB "BLACKJACK" tutorial. Will I write future articles on the care and feeding of this remarkable BASIC extension? Well, it depends on you. If you found this series useful, let your Editor know! If you have specific questions, don't hesitate to ask me or Greg (why do you suppose we published our addresses?). If you have found interesting applications, short-cuts, hints or tips, take the time o write them up and send them in for publication. Now that we've kicked off the "cheap hi-res ball," it's up to you guys and gals to catch it and run with it. Enjoy the game!